An Empirical Study on Keyword-based Web Site Clustering

Filippo Ricca, Paolo Tonella, Christian Girardi and Emanuele Pianta
ITC-irst
Centro per la Ricerca Scientifica e Tecnologica
38050 Povo (Trento), Italy
{ricca, tonella, cgirardi, pianta}@itc.it

Abstract

Web site evolution is characterized by a limited support to the understanding activities offered to the developers. In fact, design diagrams are often missing or outdated. A potentially interesting option is to reverse engineer high level views of Web sites from the content of the Web pages. Clustering is a valuable technique that can be used in this respect. Web pages can be clustered together based on the similarity of summary information about their content, represented as a list of automatically extracted keywords.

This paper presents an empirical study that was conducted to determine the meaningfulness for Web developers of clusters automatically produced from the analysis of the Web page content. Natural Language Processing (NLP) plays a central role in content analysis and keyword extraction. Thus, a second objective of the study was to assess the contribution of some shallow NLP techniques to the clustering task.

1 Introduction

The decomposition of a whole Web site into smaller cohesive units that are easier to manage during Web site evolution is often undocumented and can be only guessed (sometimes) from the directories. Over time, the navigation structure and the contents in a Web site tend to change, resulting in an increase of the complexity. The initial decomposition, if any existed at all, is often invalidated by the changes. The result is that for existing Web sites little or no support is available for a high level understanding of the overall structure.

Among the tools and techniques developed to support understanding and restructuring of existing Web sites [7, 14, 21], those based on clustering are appealing, because they offer the possibility to untangle the navigation structure, abstracting from the single pages to a view consisting of high-level, cohesive units.

Similarly to software clustering [2, 11], which aims at gathering software components into higher level groupings, thus providing the user with a more abstract, overall view of the system under analysis, Web site clustering produces a high level view of the Web site organization, in terms of clusters of pages and of the relationships among them. Such a view can be exploited in the process of Web site understanding, to gain knowledge about the organization of the entire site. More detailed information can be obtained by exploding each cluster of interest into subclusters, up to the individual pages that are the object of a change.

Given the central role of the content in Web pages, clustering methods developed specifically for Web sites can take advantage of this important source of information. In Web site clustering based on keyword extraction, described in detail elsewhere [20], the content is summarized by a list of keywords that characterize the text in each Web page. Moreover, keywords are weighted so that more specific keywords receive a higher score. Pages are considered similar if they share a large number of keywords. The contribution of each keyword to the similarity measure is proportional to their weight. Similar pages are grouped together by the clustering algorithm.

Shallow Natural Language Processing (NLP) techniques, that is techniques not based on a deep analysis of meaning, are used to determine the keywords to be associated with the text of each Web page in the site under analysis. Moreover, NLP methods can also give a weight to each keyword, according to its relevance and specificity.

In this paper, we describe the results of an empirical study that was conducted to investigate the possibility of automating Web site clustering and to assess the role of NLP. More specifically, the empirical study was designed to measure the similarity between the decomposition of a Web site produced by an expert and that obtained through keyword-based clustering. Since the latter depends on the NLP methods adopted for keyword extraction and weighting, alternative NLP methods have been contrasted to determine the best performing ones. Moreover, combinations of

NLP methods have been considered as well, with the aim of assessing whether they give complementary, composable contributions or not.

The paper is organized as follows: the next section gives a brief overview on clustering and more specifically on Web site clustering based on keywords. Some NLP methods that can be employed for keyword extraction and weighting are described in Section 3. The design of the empirical study is commented in Section 4. Experimental results are provided and discussed in Section 5. Related works are surveyed and contrasted with the present work in Section 6. Conclusions and future work are given in the last section.

2 Web site clustering

Clustering is a general technique aimed at gathering the entities that compose a system into cohesive groups (*clusters*). Given a system consisting of entities which are characterized by a vector of properties and are connected by mutual relationships, there are two main approaches to clustering [2]: the sibling link and the direct link approach. In the *sibling link* approach, entities are grouped together when they possess similar properties, while in the *direct link* approach they are grouped together when the mutual relationships form a highly interconnected sub-graph.

In the literature there exist several different clustering algorithms [22], with different properties. Hierarchical algorithms do not produce a single partition of the system. Their output is rather a tree, with the root consisting of one cluster enclosing all entities, and the leaves consisting of singleton clusters. At each intermediate level, a partition of the system is available, with the number of clusters increasing while moving downward in the tree. Divisive algorithms start from the whole system at the tree root, and then divide it into smaller clusters, attached as tree children. Alternatively, agglomerative algorithms start from singleton clusters and join them together incrementally.

When agglomerative algorithms are used, the distance measure between pairs of clusters computed at the previous step must be computed to determine which clusters to merge at the current iteration. In this respect, several rules can be followed. One of the most widely used in software clustering is the so called *complete linkage rule* [2]. This rule states that the distance measure between two already existing clusters is the minimum of the distances between pairs of enclosed elements (in turn, clusters or pages). This rule is known to privilege cohesion over coupling [2].

2.1 Clustering based on keywords

Let us assume that the keywords of each page composing a Web site have been determined. A basic keyword extraction method may just list all of the words present in a Web page and regard them as keywords. More advanced techniques might make use of Natural Language Processing (NLP) algorithms. These are described in the next section. Their output is a list of weighted keywords, associated with each Web page.

Let K be the vector of all keywords determined for a whole Web site (union of the keywords determined for each page), with each keyword uniquely represented by a single entry. A feature vector [17, 16] V_p can be built for each page p, with $V_p[i]$ holding the weight attributed to the keyword $K[i]$ in page p. A basic measure to use for the weights consists of the number of occurrences of the i-th keyword $K[i]$ in p (when a keyword is not present in a page p, the related entry $V_p[i]$ in the feature vector will be 0). More sophisticated metrics for the keyword weights (like the inverse frequency in the document) are presented in the next section.

Given the description of each Web site page in terms of a feature vector, it is possible to exploit similarity or distance measures to agglomerate entities into clusters. Similarity/distance between clusters is generalized from the similarity/distance between entities by means of the complete linkage rule (see above). In this work, we preferred a similarity measure over a distance measure, because the latter is prone to the well known problem of the sparse or empty vectors: distances become small not only when vectors are close to each other, but also when they are very sparse (or empty), thus leading to the formation of inappropriate clusters. The similarity measure used with the feature vectors described above is the normalized vector product, given by:

$$sim(p,q) = \frac{< V_p, V_q >}{||V_p|| \ \ ||V_q||} \tag{1}$$

where V_p and V_q are the feature vectors of pages p and q respectively, angular brackets indicate the scalar product, which is normalized by the product of the norms, thus giving a similarity measure which ranges from 0 to 1 (under the hypothesis of non-negative weights).

After executing the agglomerative clustering algorithm, a proper cut point is manually selected. The possibility for the user to choose a given abstraction level (number of clusters or, equivalently, cut point), and then to adjust it toward the top of the hierarchy (less clusters with more pages inside) or toward the bottom (more clusters containing fewer pages) is an important interactive facility. In fact, the right abstraction level, appropriate for the ongoing program understanding task, is typically not known a priori, and can be determined empirically by moving upward and downward in the clustering hierarchy.

Then, labels can be automatically assigned to each cluster. The keyword with the highest weight in a cluster C is assigned to C as its label.

3 Keyword extraction

Keyword-based document clustering relies on the idea that the content of a document can be characterized by a set of keywords, that is a set of terms expressing the most important concepts in the document [23]. In practice, all words in the document collection (in our case the Web site) are listed in the feature vectors, while weights (feature vector entries) measure the importance of each word in a single document (in our case a Web page).

The most simple approach to weighting the importance of a word in a document consists of measuring the number of occurrences of the word in the document. The basic intuition underlying this approach is that the most important concepts are likely to be referred to repeatedly, or, at least, more frequently than minor concepts. In the following, we will refer to this approach as the *BASE* method.

Unfortunately, there are two classes of terms that occur frequently in a text but are not useful from the point of view of the clustering task. The first class includes the so called *function words*, that is words that occur very frequently in any kind of text but do not have specific content, as for instance articles, prepositions, auxiliaries and modal verbs (the, of, is, can,...). Clearly, if two texts have both a high number of occurrences of the article "the" and the preposition "of", this piece of information should not be used to infer that they have similar content. The second class of terms includes *content words* that tend to occur uniformly in all the documents of a given collection, and so are not useful for characterizing the content of a text in contrast with other texts. This may happen when all the texts that we are trying to cluster are about the same broad topic. For instance, if all the Web pages we are processing belong to the portal of a telephone company, we may find out that the terms "telephone line" or, even worse, "telephone" occur in almost all the pages and so they are not useful to characterize the meaning of a page even if they do occur repeatedly.

To handle the problem of terms that are frequent but not useful as keywords to characterize the content of a text, at least two approaches can be used. The first technique applies the frequency-based weighting of keywords after filtering out all words included in a list of stopwords. The second technique measures the weight of terms on the basis of the inverse document frequency instead of the absolute frequency. In the rest of this section we will give more details on these two techniques, and illustrate a third technique that can be used to improve the quality of the keywords used to characterize the content of texts.

3.1 Stop words

To improve the quality of the frequency-based weighting of keywords, we can rely on a list of stopwords that should be always filtered out, as they are not relevant whatever their frequency in a document. Lists of stopwords can be produced manually and should include function words and content words that are uniformly distributed in the documents of a certain collection. As function words belong to closed classes of words, that is there are a finite and small number of articles, prepositions, etc., and given the fact that this class of words tends to occur in all kinds of text, it makes sense to include manually the list of function words in our stopword list. Instead, the list of uniformly distributed words is specific of each collection of documents, and listing them manually does not make sense, if our aim is to build a clustering algorithm working for any Web site.

Lists of stopwords can also be generated automatically. The topmost frequent words in any corpus tend to include most function words and many uniformly distributed words. Thus we could calculate the list of the most frequent words in a site and include in our stoplist the N topmost items in the list. Unfortunately such a list is prone to be both incomplete and partially wrong, first of all when the size of the site is small. Finally, there is an even simpler way of calculating a list of stopwords, based on the observation that in many languages, function words tend to be shorter than content words, whereas words with very specific meanings tend to be longer than generic words. As a consequence one can include in the stopword list all words whose length is below a certain threshold (for instance 2 or 3 characters). However, we expect the error margin of such a list to be even greater than a list based on frequency in a corpus. In the experiment presented in this paper, we used a manually produced list of stopwords including all Italian function words. However, we are planning to carry out more specific experiments to evaluate the impact of automatically built stopword lists on the clustering algorithm.

3.2 Inverse document frequency

Whereas function words can be easily handled as stopwords, the problem of uniformly distributed content words seems to be better handled by resorting to the inverse document frequency metric. As mentioned above, a content term can occur frequently in a text without characterizing the text in contrast with other texts. To avoid this problem the frequency of a term in a document should be confronted with the average frequency of that term in the whole Web site. To this aim, keywords are weighted on the basis of the Inverse Document Frequency (IDF) [12], defined as:

$$W[k] = N[k] \log \frac{D}{d[k]} \qquad (2)$$

where $N[k]$ is the absolute number of occurrences of the k-th keyword, $d[k]$ is the number of documents containing it, and D is the total number of documents.

High frequency keywords that are specific to a document ($d[k]$ small compared to D) have a high value of the inverse document frequency. On the contrary, unspecific keywords (i.e., keywords that occur uniformly in most documents) are given a small weight ($\log(D/d[k])$ is close to zero, since $d[k] \sim D$). Words that are unspecific on the basis of the IDF, can be either function or content words. Thus there is an overlap between the effect of the IDF and the effect of the stopword list as was defined in the previous section (that is as a list of function words). However consider that the IDF can fail to recognize certain function words, if for some reason they do not occur uniformly in a collection of texts. Thus it may make sense to combine IDF and stopword filtering.

3.3 Collocations

In the above presentation, we made the tacit assumption that the terms used as keywords are simple words. However, it is possible to consider lexical units that are wider than simple words (collocations). Not only can a concept be referred to with more than a word, that is a phrase, but often the concepts that characterize a text most precisely are the most specific ones, which are usually expressed with complex phrases. For this reason, as a first attempt in this direction we considered as units both single words and bigrams, that is contiguous sequences of two words.

Not all bigrams in a text should be considered as terms to our purposes. We are mainly interested in three classes of complex terms: named entities, complex lexical units, and recurrent free phrases. Named entities are terms referring to individuals, locations, organizations, and dates. Complex lexical units are the kind of multi-word expressions that can be found in dictionaries (for instance phrasal verbs such as ”put on” and idiomatic expressions such as ”roller coaster”). Finally, the notion of recurrent free phrase was introduced by [5] for a free combination of words which is recurrently used to refer to a concept. They are characterized by either high frequency in a reference corpus (e.g. ”American government”), high degree of association between words (e.g. ”first time”), or high salience (e.g. ”international summit”).

Selecting the bigrams belonging to each of these three classes is a challenging and resource demanding task. However the task can be approximated by resorting to a combination of simple statistical measures and elementary linguistic knowledge. Our strategy consists in considering as collocations the bigrams that are present in a list of frequent Italian bigrams. We built the list of frequent Italian bigrams with the following procedure: (1) we selected the topmost frequent bigrams in a reference corpus of 32 million words from an Italian newspaper; (2) we cut off all bigrams occurring less than 4 times in the reference corpus; (3) we applied a filter based on a list of function words. The aim of the filtering step is to get rid of bigrams like ”with the” or ”the only” which may occur frequently in texts but do not belong to any of the term classes mentioned above. To this extent we simply reject all bigrams including at least a word taken from a list of function words.

The recognition of collocations can be combined with both the keyword weighting metrics presented above (i.e. absolute frequency and inverse document frequency). The most relevant effect of recognizing collocations seems to be related with a rudimentary form of word sense disambiguation. Take for instance the word “line” which is highly ambiguous. Suppose that a document contains many occurrences of the expression “credit line”, whereas another document contains many occurrences of “production line”. The meaning of “line” in these two expressions is different, and the fact that both documents contain a high number of occurrences of the word “line” does not mean that they are similar. However if we can recognize that “credit line” and “production line” are lexical units, then the two expressions will be kept distinct and will not be used to infer a high similarity between the two documents. On the other side, suppose that a document contains many occurrences of “natural park”, whereas another documents includes many occurrences of “park” (in the sense of “natural park”). In this case, if we recognize “natural park” as a unit, we will lose the possibility of matching “park” and “natural park”. An experiment is needed to evaluate the real contribution of collocations to the clustering task.

4 Experimental setting

This section describes the design underlying our empirical study. A discussion of the threats to validity is given at the end of Section 5.

4.1 Experimental hypotheses

The aim of this study is answering the following research questions about Web site clustering based on keyword extraction:

Q1: *How do automatically generated clusters match the sub-sites into which an expert would decompose a whole Web site?*

Q2: *Do NLP techniques improve the performances of Web site clustering? Which NLP technique gives the largest improvement?*

Q3: *Is it possible to combine different NLP techniques to obtain an enhanced clustering method? Which combination gives the best performances?*

The first question is about the possibility to automatically group pages, based on their content, in a meaningful way with respect to the decomposition a human would adopt for a given Web site. **Q1** is a first step in the direction of assessing the usefulness of Web site clustering. If the information recovered by means of clustering is close to the information that a human would produce, its availability has the potential to simplify Web site restructuring and understanding. This is especially true if such an information is not recorded in the documentation of the Web pages or in the physical directory structure used for the site.

Questions **Q2** and **Q3** deal with the contribution of NLP to Web site clustering. Among the applicable NLP techniques, it is important to identify those that are the most effective. In fact, their integration in a clustering tool is usually non-trivial. Moreover, the combination of different techniques might give better or worse results, according to the mutual interactions they have. If they address complementary issues, their combination is potentially more powerful, while this might not be the case if they work on similar issues.

4.2 Procedure

Web sites have been selected randomly among those available on the Web. The only constraint was the presence of a non trivial amount of content in the pages (excluded Web sites are those providing a computation/service based on a user input, with almost no content attached to the involved Web pages). Considered Web sites are both *static* and *dynamic*. The latter consist of pages that are generated by server side scripts in response to the user state and interaction. Web sites have been downloaded by means of the tool *ReWeb*, which offers capabilities to explore dynamic sites as well as static ones [14].

For each downloaded Web site, an expert's decomposition (*gold standard*) was produced manually. The person who defined the gold standards for the Web sites was not part of the team that developed the clustering tool. The gold standards were revised by a second person, who submitted comments and suggestions to the expert. Together they reached an agreement on the final reference decompositions of the Web sites.

Keyword-based clustering was obtained by running a custom tool developed by some of the authors. The clustering hierarchy produced in output has been cut at all possible levels. The related clusters have been contrasted against the gold standard and the metrics described below have been obtained.

4.3 Metrics

Two metrics have been selected to address the research questions **Q1, Q2, Q3**: precision and recall. Precision and recall are defined as in [2, 6], using the notion of *intra pair*. Intra pairs are pairs of pages in a same cluster. Precision and recall are defined by comparing the intra pairs in the gold standard and those in the clustering under test:

- **Precision**: Percentage of intra pairs in the test clustering that are also in the gold standard.
- **Recall**: Percentage of intra pairs in the gold standard that are also in the test clustering.

For example, let us suppose that the expert has determined the following gold standard, out of six pages:

`{P1, P2}, {P3, P4}, {P5, P6}`

and that the clustering algorithm has produced the following result:

`{P1, P2}, {P3, P4, P5}, {P6}`

The intra pairs in the gold standard are:

`(P1, P2), (P3, P4), (P5, P6)`

while those in the test clustering are:

`(P1, P2), (P3, P4), (P4, P5), (P3, P5)`

The precision is 2/4 (only `(P1, P2)` and `(P3, P4)` are also in the gold standard) and the recall is 2/3 (`(P1, P2)` and `(P3, P4)` are in the test clustering, while `(P5, P6)` is not).

As regards question **Q1**, high values of precision and recall indicate that there is a good agreement between automatically and manually produced clusters. As regards **Q2** and **Q3**, precision and recall give two parameters useful for a detailed comparison of alternative and combined NLP methods. However, since the values of precision and recall depend on the cut point selected in the clustering hierarchy, comparing their values for different clustering techniques is not straightforward. In fact, the curves representing the successive values of precision and recall, parameterized on the cut level, should be superimposed and confronted. Only when a proper trade-off between precision and recall is chosen, a quantitatve comparison between different methods can take place. One of the most widely used way to determine the trade-off between precision and recall consists of intersecting the parameterized precision-recall curve with the identity. In other words, the point at which recall and precision are (almost) equal is selected. The problem of this approach is that the precision-recall curve might get closer to the origin at the identity, so that better trade-offs are available at other points. For this reason, we have decided to select a point in the precision-recall curve which lies within a given band around the identity and at the same time maximizes the sum of precision and recall:

$$\begin{cases} |p-r| \leq \epsilon \\ (p^*, r^*) = argmax(p+r) \end{cases}$$

Among the values of p (precision) and r (recall) that are inside the lines: $p - r = \epsilon$ and $r - p = \epsilon$ (with $\epsilon = 0.3$ in our setting), those that maximize the sum $p + r$ are selected.

With such a choice, different method can be compared by contrasting the respective values of *avg* = $(p^* + r^*)/2$.

5 Experimental results

Table 1 contains some descriptive information about the ten Web sites selected for the study: number of pages downloaded by **ReWeb**, actual number of pages analyzed, number of first-level directories. The last two columns show the number of clusters in the manually produced gold standard and the number of clusters in the best clustering produced by the best-performing algorithm. The difference between number of downloaded pages and number of analyzed pages is due to a preprocessing step, that has been executed to delete non-HTML pages (images, audio files, etc.), error pages and void pages. Moreover, pages in a language different from Italian have been also removed, since the NLP techniques integrated in our clustering tool work only on Italian documents. Language recognition was achieved by means of the automated technique presented in [19].

Table 2 shows the best values of precision (p^*) and recall (r^*) produced by each technique considered in this study (*avg* gives the average between p^* and r^*). These values represent the optimal trade-off between precision and recall, according to the criteria described in Section 4. The column BASE gives precision and recall, obtained using the simplest approach, where keywords are weighted by the number of occurrences in each page. IDF has been obtained by applying the inverse document frequency formula, while the column STOP shows the results obtained by deleting the stop words from the set of all the keywords. STOP+COLL and IDF+COLL refine the base methods, by considering also groups of two words (collocations) as possible keywords. The last column of the table combines STOP and IDF.

For each Web site, the highest value of *avg* among the different techniques is in boldface. Usually, the technique with the highest *avg* is also the one which gives the best precision-recall curve, according to a visual inspection. For example, let us consider the plot in Figure 1 for the Web site *pescare*. The best curve (dashed) is clearly STOP, and the same technique gives also the maximum value of *avg*.

5.1 Research question Q1

In general, the results of content-based automated clustering look good. In seven cases (out of ten), the best clusters (best technique and best clustering) are close to the gold standard, having a value of *avg* greater than 70%.

The worst case is `www.naturalia.org`, where *avg* is slightly below 30%. *Naturalia* is a mono-thematic Web site that promotes national parks in Europe. Our gold standard follows the decomposition of the site into directories, where pages are divided by nation. Our clustering techniques do not succeed separating pages by nation, because often pages with a similar content belong to different nations (for example, `italia/aree.htm` and `germania/aree.htm`) and are consequently grouped together by our techniques. However, the decomposition proposed by our algorithms is meaningful and is potentially useful, though different from the gold standard. The grouping is by themes (animals, lakes, areas, etc.) and not by nations. It represents another compositional view of the Web site.

5.2 Research question Q2

NLP techniques improve substantially the performances of Web site clustering. In nine cases (out of ten), shallow NLP techniques give results that are better than those obtained with the simplest approach (BASE). See for example Figure 2, where all curves associated with the usage of NLP are superior than the curve BASE.

Let us focus on the two *basic* NLP techniques, i.e., STOP and IDF. We will consider their combination and their extension by means of the collocations (COLL) in the next sub-section about question **Q3**. Between STOP and IDF, the technique that gives the best results is STOP (this is pretty clear in Figures 1 and 2).

STOP improves BASE in nine cases out of ten. In four cases STOP gives the highest *avg* and in two cases it gives values slightly below STOP+COLL

IDF behaves worse than STOP. It exceeds BASE in only three cases and it is the best curve in just one case. Moreover, in four cases this technique gives worse results than BASE.

5.3 Research question Q3

Collocations seldom improve the base techniques (IDF and STOP), and when this happens, the improvement is minimal (except for one case, the Web site *assfor* with the technique IDF). In addition, the combination IDF+STOP does not improve, in a significant way, the base techniques taken singularly and sometimes it makes them worse.

Collocations seem not much useful. Let us consider the curves for the Web site *pescare* (Figure 1) and *promoturpejo*, (Figure 2). The curve STOP+COLLOC. and IDF+COLLOC. are respectively below STOP and IDF (in the interesting region, which is around the identity).

Web site	Downloaded Pages	Analyzed pages	Directories	Gold standard clusters	Best algorithm clusters
www.valmalencofree.com	339	178	64	24	32
www.naturalia.org	715	210	7	17	34
www.agriturismoitaly.it	1247	399	15	9	21
www.assfor.it	338	335	7	14	24
pescaonline.awsolutions.it	188	180	19	12	16
utenti.lycos.it/pescare	522	316	18	19	26
www.promoturpejo.it	234	167	9	10	12
www.certosaviaggi.it	318	221	43	18	20
www.lessiniapark.it	243	212	26	16	15
www.alpiapuane.com	519	450	19	19	21

Table 1. *Analyzed Web sites.*

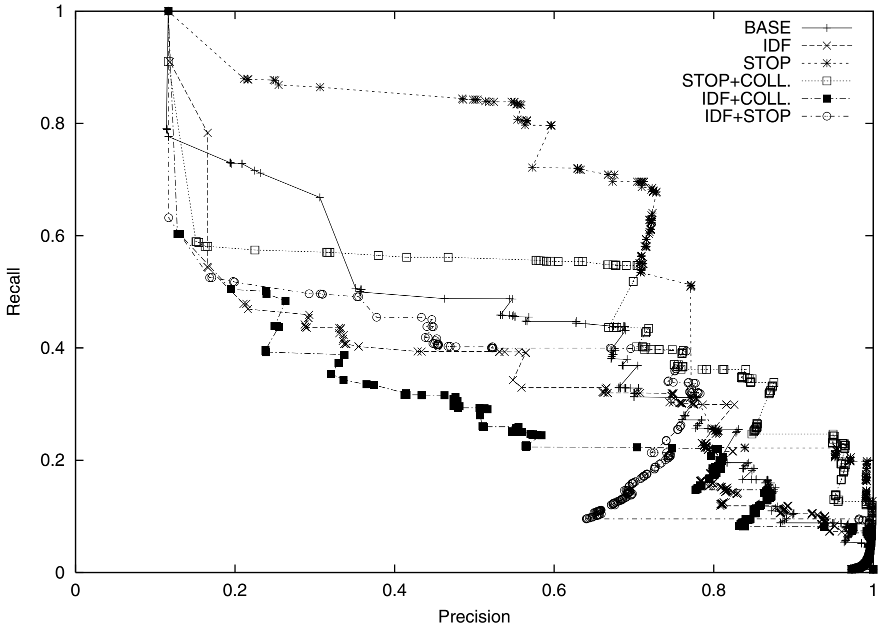

Figure 1. *Precision-recall plot for the Web site* `pescare`.

If we consider the optimal point selected in the precision-recall plot, contradictory facts emerge. In fact, in four cases (out of ten) collocations (STOP+COLL) improve the STOP technique, but in six cases they make it worse. In four cases collocations (IDF+COLL) improve the IDF technique, but in six cases they worsen it. In three cases IDF+COLL is worse than BASE and in two cases STOP+COLL is worse than BASE.

In general, combining IDF and STOP do not improve the results. Only in two cases (Web sites: *valmalencofree* and *naturalia*) it happens that IDF+STOP gives the best values. From the analysis of the results, it seems that only when BASE is very negative (i.e. `avg` $< 30\%$) IDF is better than STOP and IDF+STOP gives a significant improvement.

5.4 Discussion

In summary, our study indicates that the clusters produced automatically are close to those that a human would produce for a given Web site, and that NLP techniques give a substantial contribution to the production of good clusters based on Web page contents. Overall, the best NLP technique to use consists of removing the stopwords from the text of the Web pages. Weighting the words in a page according to their specificity (IDF) gives inferior results. The extension of the basic NLP techniques with the identification of the collocations (STOP+COLL and IDF+COLL) and the combination of the two basic techniques (STOP+IDF) do not represent a significant improvement, in that they give slightly superior, but sometimes in-

Web site	BASE $p^* r^*$ avg	IDF $p^* r^*$ avg	STOP $p^* r^*$ avg	STOP+COLL $p^* r^*$ avg	IDF+COLL $p^* r^*$ avg	IDF+STOP $p^* r^*$ avg
valmalencofree	8.3 33.0 20.6	60.6 33.0 46.8	51.5 24.6 38.1	47.2 22.9 35.1	56.4 31.5 43.9	**60.6 33.8 47.2**
naturalia	10.6 33.8 22.2	39.6 12.3 25.9	9.0 37.9 23.4	9.1 38.1 23.6	43.5 14.2 28.8	**42.0 15.8 28.9**
agriturismoitaly	40.3 61.3 50.8	25.2 43.6 34.4	**37.5 67.5 52.5**	31.9 56.1 44.0	24.3 45.4 34.8	24.0 49.8 36.9
assfor	45.2 59.0 50.2	62.9 51.8 57.3	**87.8 61.8 74.8**	80.6 66.9 73.7	83.9 57.6 70.7	80.2 62.9 71.5
pescaonline	67.8 41.4 54.6	**84.9 57.2 71.1**	80.6 54.9 67.7	83.6 55.6 69.6	76.9 55.2 66.1	73.8 50.0 61.9
pescare	68.8 43.8 56.3	56.5 39.2 47.8	**71.3 69.6 70.4**	70.9 54.7 62.8	55.5 25.9 40.7	69.5 39.9 54.7
promoturpejo	54.7 64.6 59.6	89.2 67.7 78.4	**91.7 76.9 84.3**	78.6 78.9 78.8	88.2 65.4 76.8	80.2 68.3 74.2
certosaviaggi	85.3 55.4 70.3	76.2 72.5 74.3	81.0 75.5 78.2	**81.6 75.4 78.5**	70.1 40.4 55.2	87.1 60.5 73.8
lessiniapark	69.2 87.7 78.4	76.4 76.6 76.5	87.2 88.2 87.7	**86.1 91.2 88.6**	68.7 95.3 82.0	78.2 77.0 77.6
alpiapuane	**86.8 86.3 86.5**	81.8 90.8 86.3	60.0 30.6 45.3	62.8 34.3 48.6	39.5 67.9 53.6	71.1 43.2 57.1

Table 2. *Precision and recall for the analyzed Web sites.*

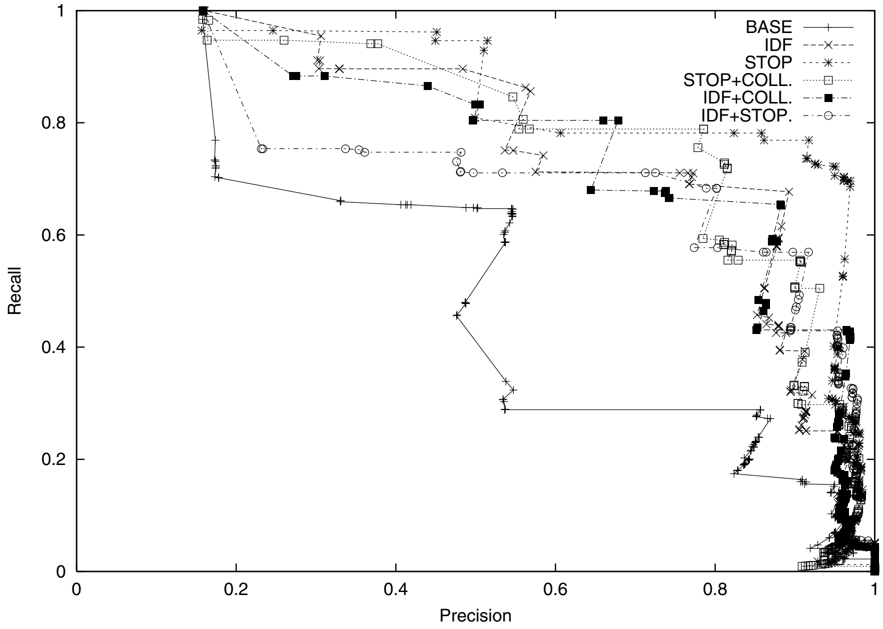

Figure 2. *Precision-recall plot for the Web site* `promoturpejo`.

ferior, results.

The two basic NLP technique STOP and IDF go in a similar direction (filtering irrelevant information), but with different means. The most effective way to filter irrelevant information seems to be represented by the removal of the stopwords. Moreover, the combination of IDF and STOP does not improve the basic results because these two techniques perform a similar task, though in different ways. In other words they are not complementary techniques, and this might explain the low performances of their combination.

Collocations are potentially useful, in that they could identify specific groups of words that characterize the content. However, they are also potentially dangerous, when they reduce the weight of an important word which sometimes appear in a collocation. The results we obtained are contradictory, but the positive effects seem not to be particularly relevant.

5.5 Threats to validity

The main causes that might limit the generalization of our results to other Web sites are analyzed in this section.

The *size* of the considered Web sites is quite typical, when the purpose of the site is providing some information (sites have been sampled randomly). However, Web sites with a size that is one or more orders of magnitude larger are expected to be non-typical, but could exist. The behavior of automated clustering in such cases cannot be easily extrapolated from what was observed in our cases.

Web sites that have a very limited content, in that they provide a computation or a service (e.g., a train time table) can be hardly clustered by means of keyword extraction, in that the starting information (pages with several content words) is missing. Thus, the method applies to Web sites that are known to be mainly organized around contents. It should be noted that no assumption is made about the way contents are generated. The case of Web pages that are produced dynamically by content management systems or by scripts that access a database, are perfectly compatible with our approach.

The representativeness of the Web sites with respect to the possible variability of the contents is also questionable. In fact, the topics that are dealt with in existing Web sites vary on a very large spectrum. It is practically impossible to sample uniformly such a spectrum. It might be the case that some contents can be clustered more easily than others. For example, we expect that a Web site concerning mathematics and containing a lot of mathematical formulas and little text could represent a difficult or even impossible task for our clustering algorithm.

Finally, one central question our study does not address is about the usefulness of clustering for Web site restructuring and understanding. The capability to generate a decomposition of a Web site which is close to one produced manually and which is meaningful to a human is a good premise, but it is not sufficient to demonstrate the actual usefulness of the method in the execution of real maintenance tasks. This could be the subject for another empirical study.

6 Related work

Clustering has several uses in program understanding and software reengineering [2, 11, 22], and has been recently applied to Web applications [7, 8, 15]. NLP and information retrieval methods have been applied to software engineering with the aim of recovering traceability links [3, 4, 13], of supporting program comprehension [1, 10] and code reuse [9].

In [7] an approach to support the comprehension of Web applications by exploiting clustering techniques has been proposed. The approach is based on a conceptual model of a Web application and on a similarity measure between components that takes into account both the type and the topology of the links. This measure is exploited by a hierarchical clustering algorithm which produces a hierarchy of system partitions. Download and comprehension of the considered Web applications are conducted using the reverse engineering tool WARE.

In [8] an approach to identify duplicated pages (i.e., clones) in a Web application is proposed. Two different methods based on different similarity measures have been defined and experimented with: one exploiting the edit distance and the other one based on the frequency of the HTML tags in a page. The underlying descriptive feature (the HTML structure of a page) can be used as a further basic feature for clustering. This feature was actually exploited in [15], where an approach is presented for the identification of Web pages that can be migrated to a dynamic version, in that they share a similar structure.

An approach to clustering documents based on keyword extraction is described in [18]. Whereas in our approach the separation of documents belonging to different languages is carried out with an independent algorithm, before applying document clustering, the authors in [18] apply their clustering algorithm to a multilingual collection of documents and obtain as a product of the clustering a grouping of documents of the same language. Their approach is crucially based on multiword keywords (collocations), which are extracted from the document collection itself resorting to quite sophisticated statistical techiques. However the clustering algorithm has been evaluated on one document collection only, and no comparison is provided with other keyword based clustering approaches.

This work extends our previous work [20], where Web site clustering based on keyword extraction was presented for the first time. The main contribution of the present work with respect to [20] and the other relevant literature consists of the empirical study that was conducted to answer a very fundamental question about the similarity between automatic decomposition and expert's decomposition. This is the first work that addresses such a question for Web sites in a systematic way. Moreover, the role of NLP in Web site clustering has never been assessed before by means of a controlled experiment.

7 Conclusions and future work

Web site clustering is potentially a useful technique in support to program understanding during Web site evolution. This work represents a first empirical study aimed at evaluating such a method by means of a controlled experiment, with cases taken from the real world.

The first question we addressed is about the difference between an automatically produced decomposition and one that is generated manually by an expert. In this respect, results are positive. If Web pages are clustered according to their content, properly summarized by a list of keywords, the resulting clusters are meaningful and close to those a human would produce.

Since keyword-based clustering depends on the ability to process the content of the Web pages, we investigated the alternative NLP techniques that can be used for such a purpose. Our result is that the best results are obtained if irrelevant words are filtered (stopwords) and the remaining words are simply weighted by counting the number of occurrences.

Generalization from a single study is always difficult.

We identified some main threats to validity which could be considered in successive studies. For example, a study on Web sites of larger size could be conducted. Ways to better sample along the dimension of the content variability could be devised. Moreover, the fact that a view extracted from the source code is meaningful does not imply that it is also useful. Studies on the usefulness of clustering – in the field of Web applications, but also in the wider field of generic software systems – are lacking, and future work should be definitely devoted to them. Finally, clustering techniques that are not based on the content could be subjected to a similar assessment as we did for keyword-based clustering.

References

[1] N. Anquetil and T. Lethbridge. Assessing the relevance of identifier names in a legacy software system. In *Proceedings of CASCON*, pages 213–222, December 1998.

[2] N. Anquetil and T. C. Lethbridge. Experiments with clustering as a software remodularization method. In *Proc. of the 6th Working Conference on Reverse Engineering (WCRE'99)*, pages 235–255, Atlanta, Georgia, USA, October 1999. IEEE Computer Society.

[3] G. Antoniol, G. Canfora, G. Casazza, and A. D. Lucia. Information retrieval models for recovering traceability links between code and documentation. In *Proceedings of the International Conference on Software Maintenance (ICSM)*, pages 40–51, San Jose, CA, USA, October 2000. IEEE Computer Society.

[4] G. Antoniol, G. Canfora, A. D. Lucia, and E. Merlo. Recovering code to documentation links in oo systems. In *Proceedings of the 6th Working Conference on Reverse Engineering (WCRE)*, pages 136–144. IEEE Computer Society, October 1999.

[5] L. Bentivogli and E. Pianta. Beyond lexical units: Enriching wordnets with phrasets. In *Proceedings of the Research Note Sessions of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL'03)*, pages 67–70, Budapest, Hungary, April 2003.

[6] J. Davey and E. Burd. Evaluating the suitability of data clustering for software remodularization. In *Proc. of the Seventh Working Conference on Reverse Engineering (WCRE'00)*, pages 268–277, Brisbane, Australia, November 2000. IEEE Computer Society.

[7] G. A. D. Lucca, A. R. Fasolino, U. D. Carlini, F. Pace, and P. Tramontana. Comprehending web applications by a clustering based approach. In *Proc. of the 10th International Workshop on Program Comprehension (IWPC)*, pages 261–270, Paris, France, June 2002. IEEE Computer Society.

[8] G. A. D. Lucca, M. D. Penta, and A. R. Fasolino. An approach to identify duplicated web pages. In *Proc. of the 26th Annual International Computer Software and Applications Conference (COMPSAC)*, pages 481–486, Oxford, England, August 2002. IEEE Computer Society.

[9] Y. S. Maarek, D. M. Berry, and G. E. Kaiser. An information retrieval approach for automatically constructing software libraries. *IEEE Transactions on Software Engineering*, 17(8):800–813, 1991.

[10] J. Maletic and A. Marcus. Supporting program comprehension using semantic and structural information. In *Proceedings of the 23rd International Conference on Software Engineering (ICSE)*, pages 103–112, Toronto, Canada, May 2001. IEEE Computer Society.

[11] S. Mancoridis, B. S. Mitchell, Y. Chen, and E. R. Gansner. Using automatic clustering to produce high-level system organizations of source code. In *Proc. of the International Workshop on Program Comprehension*, pages 45–52, Ischia, Italy, 1998.

[12] C. D. Manning and H. Schtze. *Foundations of Statistical Natural Language Processing*. The Mit Press, Cambridge MA, 1999.

[13] A. Marcus and J. Maletic. Recovering documentation-to-source-code traceability links using latent semantic indexing. In *Proceedings of the 25th International Conference on Software Engineering (ICSE)*, pages 124–135, Portland, OR, USA, May 2003. IEEE Computer Society.

[14] F. Ricca and P. Tonella. Analysis and testing of web applications. In *Proc. of ICSE 2001, International Conference on Software Engineering, Toronto, Ontario, Canada, May 12-19*, pages 25–34, 2001.

[15] F. Ricca and P. Tonella. Using clustering to support the migration from static to dynamic web pages. In *Proc. of the International Workshop on Program Comprehension (IWPC)*, pages 207–216, Portland, Oregon, USA, May 2003. IEEE Computer Society.

[16] G. Salton. *Automatic Text Processing: The Transformation Analysis and Retrieval of Information by Computer*. Addison-Wesley Publishing Company, Reading, MA, 1989.

[17] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw Hill, New York, 1983.

[18] J. F. Silva, J. Mexia, A. Coelho, and G. P. Lopes. Multilingual document clustering, cluster topic extraction and data transformation. *Lecture Notes in Artificial Intelligence (Progress in Artificial Intelligence)*, 2258:74–87, 2001.

[19] P. Tonella, F. Ricca, E. Pianta, and C. Girardi. Restructuring multilingual web sites. In *Proc. of the International Conference on Software Maintenance (ICSM 2002)*, pages 290–299, Montreal, Canada, October 2002. IEEE Computer Society Press.

[20] P. Tonella, F. Ricca, E. Pianta, and C. Girardi. Using keyword extraction for web site clustering. In *Proc. of the International Workshop on Web Site Evolution (WSE 2003)*, pages 41–48, Amsterdam, The Netherlands, September 2003. IEEE Computer Society Press.

[21] P. Warren, C. Boldyreff, and M. Munro. The evolution of websites. In *Proc. of the International Workshop on Program Comprehension*, pages 178–185, Pittsburgh, PA, USA, May 1999.

[22] T. Wiggerts. Using clustering algorithms in legacy systems remodularization. In *Proc. of the 4th Working Conference on Reverse Engineering (WCRE)*, pages 33–43. IEEE Computer Society, 1997.

[23] P. Willet. Recent trends in hierarchic document clustering: a critical review. *Information Processing and Management*, 24(5):577–597, 1988.