EPiC Series in Language and Linguistics

Volume 1, 2016, Pages 274–282

CILC2016. 8th International
Conference on Corpus Linguistics

Review web pages collector tool for thematic corpus creation

Lisa Medrouk, Anna Pappa, and Jugurtha Hallou

[1] Paris 8 University, LIASD, Saint- Denis, France
lmedrouk@ai.univ-paris8.fr
[2] Paris 8 University, LIASD, Saint- Denis, France
ap@ai.univ-paris8.fr>
[3] Paris 8 University, LIASD, Saint- Denis, France
jhallou@ai.univ-paris8.fr>

Abstract

We present a method of automaticaly extracting and gathering specific data text from web pages, creating a thematic corpus of reviews for opinion mining and sentiment analysis. The internet is an immense source of machine-readable texts [11] suitable for linguistic corpus studies[3][1]. Though, specific tools of web information extraction research domain as well as from the NLP do not include an open source system able to provide a thematic corpus according to an end-user request[16].
The need of use natural texts as databank for opinion mining and sentiment analysis is increased since the expansion of the digital interaction between users and blogs, forums and social networks.
The RevScrap system is designed to provide an intuitive, easy-to-use interface able to extract specific information from accurate web pages returned by search engine's request and create a corpus composed by comments, reviews, opinions, as expressed by users' experience and feedback. The corpus is well structured in xml documents, reflected Singler's design criteria[4].

1 Introduction

The Web constitutes an ubiquitous, inescapable data collection source, providing a mine of inexhaustible informations in various languages and types. The interesting point for our method about creating corpora from the web resides on the fact that internet data could enrich existing corpora as well as giving accessibility to less viewed pages. Web pages are composed by large amount of unrelated material like navigational elements, advertisssements, and templates[9] making the web collecting process a complex procedure, thus, creating a clean thematic corpus from the web, is still a challenging task[6]. The main objective of our research is to automatically create a thematic corpus from specific text data needed to develop further opinion analysis tools. The concept of the tool is designed as a self-contained web application that could be deployed to any java web server. Before proceed with an explanation of our approach it will be helpful to define the problem and outline aspect and semantic assumptions. To simplify the problem we present the construction of our system by steps beginning by some definitions and terms

disambiguations largely used in NLP domain, in order to set the theoretical background of our research.

- Definition 1 (*thematic corpus*) : for our system a thematic corpus is a collection of raw text data gathered from web pages relatives to specific keywords request.
- Definition 2 (*keywords*) : the instance keyword used for the search engine could be viewed as a tuple composed by a synonym of reviews, comments, opinions followed by the term (product, item, person, place or other) for which we want to gather information for further analysis. The keyword has the same form in any language asked.
- Definition 3 (*aspect-result*) : the result presented in the final xml opinion document is an aspect instance, viewed also as a tuple composed by the url and the keyword used in the request.

The information provided by the web is complicated to extract due to the increasing complexity of the page layouts with menus, forms, sidebars, advertisements and all the unrelated material present in the pages. The RevScrap system is a tool able to consequently grap specific text data from website pages returned via web index to users requests. The applied method can be viewed as a set of sequential steps of process, each one giving access to user. The first step of the method allows the user to set the keywords which are the input to search engine request, for example return opinions + item. The second step is gathering URL(s) filtered by a parser excluding "noise" such as advertising and pages of formatting documents (pdf or ppt). The third step is the main process : it uses a specific DOM tree algorithm named ScrapRev for the detection and the identification of the comments nodes regardless the heterogeneity of the page structure and only keeping the most useful texts avoiding redundancy and noise. The ScrapRev technique is based on the top down tree-parsing algorithm, which it's easier to implement by manual programming. The results are collected to well structured xml documents named after the request's terms. The size of the xml documents varies on the quantity of comments present in the retrieved web page.

Constructing specific web corpora using keywords via search engine queries is a prolific domain but we still lack a thematic reviews corpus ready to use for opinion mining, in this paper we will show that despite the heterogeneity and the complexity of the web pages, our targeted datas are mainly always located in the same repeated nodes regardless languages, allowing us to have a salient feature to automate this distinction and provide a ready to use thematic corpus.

2 Related work

Actual NLP tools offer a great deal of concordancers, search engines and text-analyzers, like BootCat [2], Webcorp Live [7] or the Linguist's Search Engine (LSE) [14] but they either use previously mostly hand-made corpora, or do not provide an easy downloadable results suitable for further use, or simply are much too expensive for academic budgets.
The methods of information extraction (IE) applied to web information services, offer effective and efficient technologies to discover valuable and relevant knowledge in predefined set of concepts expressed by a corpus of texts put together mostly manually to form a specified information domain[15]. Building specific Web corpora using automated search engine queries started in 2001 with CorpusBuilder of [5] a tool for automatically generating Web-search queries tool to construct corpora for minority languages. Baroni [2] proposed a method for building domain specific Web corpora named BootCaT, using a characteristic domain small set of seed

for automated queries, the results are then used to extend the corpus and so forth.[8] used the same method to built a corpus factory of major world languages, by gathering a seed word list of hundred words and randomly select three of these words to create a search query, repeating several thousand times the operation in order to obtain a large corpus, the retrieved matching pages are then cleaned, tokenized, lemmatized, part of speech tagged before being load as the resulting corpus. Sharoff [16] also used the same approach with a 500 word seeds list.
Our research is targeted toward automatically creating thematic corpus, and is closer to previous works that showed that a web page can be partitioned into multiple blocks and prooving that eliminating irrelevant blocks from pages can facilitate data accessibility and data mining using DOM (Document Object Model). In our work the page partitioning provides the necessary elements of parsing and gathering relevant nodes for opinion mining. Noise detection based on DOM as level features of segments is already used by Yi [10] who introduced a simplified DOM data structure and a style tree to detect and eliminate noisy information in web pages and used web page clustering and classification as evaluation method showing that the method boosted dramaticaly data mining dataset. Menad [12] used HTML hierarchy architecure and Levenshtein distance of retreived web pages to detect relevant data and exclude irrelevant content. Lin [17] also tries to partition a Web page into blocks and identify content ones. Finally, BlogBuster [13], a tool for extracting a corpus from the blogosphere uses an engine that collect extracted required information from DOM nodes too.

3 The proposed technique architecture

Our thematic extraction algorithm is based on the following observation : The Web pages are segmented into atomic text blocks, while the review/comments in web pages follow different types of layout. We observed text types like boilerplate, content, as well as other classes like headline, user comments etc. We noticed that comments are mainly in specific redundant nodes. Therefore, we based the system on a specific DOM tree algorithm named DOMScrapRev for the detection and the identification of the comments nodes regardless the heterogeneity of the page structure and only keeping the most useful texts, excluding redundancy and keeping the most valuable reviews.

3.1 The system RevScrap

At a first place, our approach intend to provide a user-friendly and minimal effort environment. In the first step we define a set of keywords : a set of terms semantically close to opinion (in our case), which are the input to search engine request, for example in french a keyword could be avis utilisateurs + item. For our tests, we defined equivalent keywords in three languages : english, french and greek and we used the google search engine and saved all the matching returned URLs.
This lexical cross by using synonyms in the request avoids to get URLs only with a description of the item, without users reviews. A first parsing is taking place once the request is treated and the search engine returns matching URLs. A filter is applied to remove advertising or formatting type of pages for instance pdf or ppt format. This reduction is necessary to avoid useless or irrelevant links. Each returned page is processed by the RevSCrap DOM top down tree-parsing algorithm. By processed we mean partitioning the page in several content blocks according to W3Cs DOM architecture. Based on DOM (Document Object Model) which identify the logical structure of documents and the way it's handled in term of access and manipulation. In a DOM, a Web page can be parsed and represented with a tree structure which models the

parent-child relations between HTML tags, and in which leaf nodes contains content or anchor texts. Fig.1 shows an example of a graphical representation of a DOM tree.

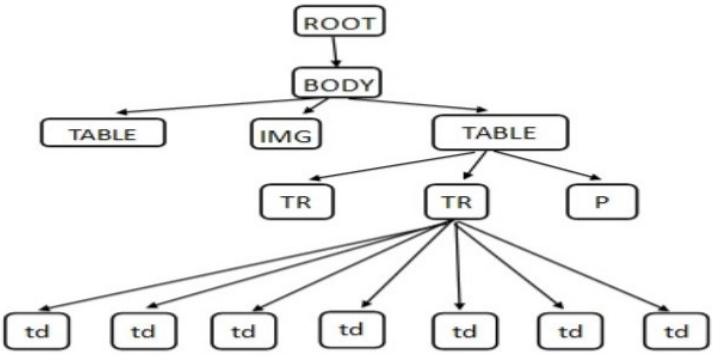

Figure 1: A graphical representation of a DOM tree example

Our approach is based on the assumption that in a given HTML page divided in blocks, the comments are based in the repeated nodes of a sub block. Thus we process the page looking for the repeated nodes. the content of the nodes that are repeated more than six times with an average content superior to 250 characters are set as possible review content to be retrieved. A less important number of characters may lead to retrieve irrelevant content as links, images, etc. All the content of repeated leafs nodes are marked as possible reviews and saved for further processing after reduction of the potential appropriate nodes. We present the module that deals with the problem of filtering the web pages and summarizes the recursive nodes in the next section.

3.2 Filters

This module deals with the reduction of "noise" of the potential nodes containing relevant information to our request text data. Web Reviews are usually associated with scores, stars evaluation and most likely dates. The first tests performed on the filters tend to show that the Date node is the most common node associated with reviews. Therefore we set the Date node as the first filter associated with the matching retrieved comments, the algorithm jump up to the third ancestor looking for a date Node.
A second filter is associated to the comments that have an URL who doesn't completely match the seed search word. For this purpose we proceed with a pattern matching parsing and a query matching to all the retrieved comments with non matching URL. For example, for a query of type : "avis samsung galaxy S4" it is possible to retrieve comments associated with an URL of a galaxy S5 as shown in the figure 2 Thus, this second pattern matching filter is set to avoid these mismatch cases.

In the figure 3, we used the same keyword query " samsung galaxy S4" in greek, another UTF-8 language, and as we expected, some URLs returned ok and some mismatched. We applied the filter in order to avoid activating mismatched web pages

<idCommentaire>9ed38bfe-d858-45fe-ad49-5d0bf831e4f0</idCommentaire>
<contenu>bien date :31 mars 2014 vous aimez : qualité photo, qualité écran tres beau telephone + 15points 15of 15voted this as helpful. faîtes connaître produit : merci! vous avez réussi à soumettre un commentaire pour cet avis.
</contenu>
<score>0.0</score>
<domaine>avis.orange.fr</domaine>
<url>http://avis.orange.fr/6044-fr_fr/3561292201638/samsung-samsung-galaxy-s5-noir-reviews/reviews.htm</url>

Figure 2: A URL mismatch with a seed word

query :

"γνώμες καταναλωτών για samsung galaxy S4"

links returned ok:

- http://www.gameover.gr/news/item/27512-diefkriniseis-gia-ta-afthentika-samsung-galaxy-s4/27512-diefkriniseis-gia-ta-afthentika-samsung-galaxy-s4

- http://www.myphone.gr/forum/showthread.php?t=390872

- http://www.didymoteicho.net/forum/49/14236-samsung-----galaxy-s4---html

link returned mismatched :

- http://techblog.gr/tag/samsung-galaxy-s5/

Figure 3: Example of query in Greek and some returned URLs

In addition to the previous listed filters, the system identifies the review URLs with multiples review pages and sink in theses pages to retrieve all the comments associated with this pages.

3.3 The overall algorithm

The overall algorithm is as follow :

```
1: Set<String> queryResults = getGoogleResults(request);
2: for(String result : queryResults)
3:              urlList = crawl(result);
4: end for
5: for(URL url : urlList)
6:              treeDOM = treeDOMConstructor(url);
7:              listComments = commentsSearch(treeDOM);
8: end for
```

Algorithm 1: OverallAlgorithm

```
1: nbrRotate = 2;
2: while (nbrRotate > 0)
3:              htmlDoc = Jsoup.get();
4:              links = getLinksAbsolutes(htmlDoc.body);
5:              links = getLinksRelatives(htmlDoc.body);
6:              noiseFilter(links);
7: end while
```

Algorithm 2: Crawl

```
1: htmlDoc = Jsoup.get();
2: Elements nodes = body.children();
3: level = 1;
4: While (nodes != null)
5:              for (Node node : nodes)
6:                      treeDOM.add(node, level);
7:              end for
8: end While
```

Algorithm 3: treeDOMConstructor

```
1: if(treeDOM != null)
2:  htmlDoc = Jsoup.get();
3:  Elements nodes = body.children();
4:  for(int level : levels)
5:   get all nodes with the level
6:   if(number of upper than 5)
7:    get text of all nodes;
8:    if(medium text lenght of nodes > 300)
9:     listeNodesOfComments.add(nodes);
10:   end if
11:  end if
12: end for
13: dateFilter(nodes.text);
14:end if;
```

Algorithm 4: commentsSearch

3.4 Aspect - Results

The resulting output is a collection of xml files with the following tags : idcomment, content, score, domain, path and URL, as shown in the following figures :

```
<commentaire>
<idCommentaire>ae8a1dd8-dfc1-4265-a214-7855ad8f08a0</idCommentaire>
<contenu>maximei membre junior inscrit: 5 septembre 2014 messages:
34 j'aime reçus: 1 en ayant l'iphone 6 depuis sa sortie, je peux
t'assurer que l'autonomie a vraiment été améliorée. je tiens
facilement 1 jours et demi entre deux recharges avec plus de 15h en
veille et 8h en utilisation. après ça dépendra aussi de ton
utilisation et de tes réglages (4g activée, actualisation de la
météo en arrière plan et wifi allumé la moitié du temps pour moi).
#2 maximei, 20 octobre 2014</contenu>
<score>0.0</score>
<domaine>forums.macg.co</domaine>
<url>http://forums.macg.co/threads/autonomie-iphone-6.1253720/</url>
<path>div.uix_message </path>
<nbrMotsListeMotsUnGram>0</nbrMotsListeMotsUnGram>
</commentaire>
```

Figure 4: Resulting Xml files format

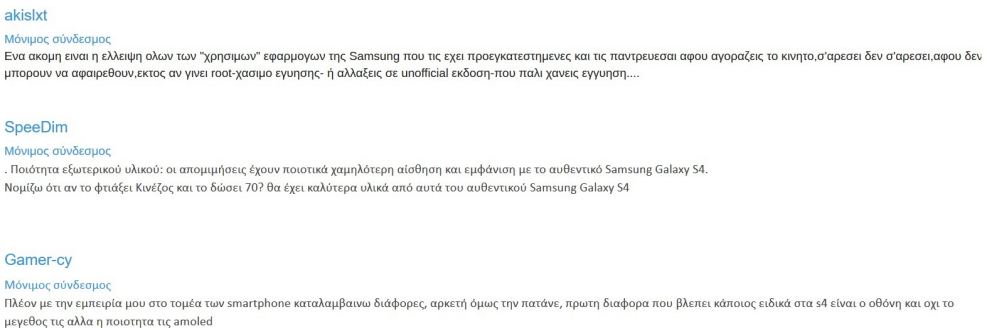

Figure 5: Greek results

4 Conclusion

The goal of our work is to automatically create a user-friendly system able to build review corpora, in different languages, according to user's specific thematic query. The system provides an interactive interface working sequentially based on a step by step method. First of all we set the terms standing for the opinion mining keywords. A small lexicon of synonyms is then created that will be used in search engine query in order to widely cover the query's semantic field.

The search engine returns many URLs and a first filter is applied to remove advertisement and format type web pages like pdf. The links are activated after parsing and filtering the relevant html nodes. The nodes of opinion data are detected and parsed. The data is then gathered to xml documents highly reduced from noises. Thus, the first test batch was set to compare the RevSCrap system to a hand made one, both performing the same query search in order to evaluate the RevScrap capacity in retrieving only relevant comments, the idea was to query the same product review and compare the results, the selected queries are "avis Samsung S4" and "avis iphone 6". For tests purposes we set a test threshold fixed at the first 100 returned URLs.
The following tables showed the resulting results :

Samsung S4	**Relevant URLs**	**Non relevant URLs**	**Error**	**Retrieved Comments**	**Relevant Comments**
Hand made results	57	40	3	885	418
RevSCrap results	72	23	5	736	666

iphone 6	**Relevant URLs**	**Non relevant URLs**	**Error**	**Retrieved Comments**	**Relevant Comments**
Person results	49	48	3	593	335
RevSCrap results	75	20	5	717	671

Table 1: RevSCrap versus hand made corpus

According to this experiments, we can state that RevScrap returned greater results regarding, the number of relevant retrieved comments and relevant URLs. We can achieve high link coverage without a greedy time algorithm.

5 Future Work

In this paper, we have presented a simple, yet effective system for automatically generating thematic review using an interactive web interface. We have shown that our method perform well in retrieving and cleaning review web pages providing ready to use review corpus. While our current results indicate a good performance it can be improved in several aspects.

1. Filtering on the Date node was the most sensible choice but for a better accuracy it will suitable to add more node filters like the Score and or star nodes.
2. Testing the extension of the system for supporting more languages, the system has already been updated with two news languages which are English and Greek.
3. Further enhancements like using ontologies to extend the comments base is discussed and might be a solution for adding more value for sentiment analysis research. Future work involves comparison the accuracy and the computational efficiency of aspect extraction and gathering method with other approaches. We are testing multiple threads in a parallel processing version. In addition, larger scale quantitative evaluation of our opinion data corpora will be conducted.

References

[1] Kilgarriff Adam and Grefenstette Gregory. Introduction to the special issue on the web as corpus. *Comput. Linguist.*, 29(3):333–347, September 2003.

[2] Marco Baroni and Silvia Bernardini. Bootcat: Bootstrapping corpora and terms from the web. In *In Proceedings of LREC 2004*, pages 1313–1316, 2004.

[3] William H. Fletcher. Making the web more useful as a source for linguistic corpora. In *Corpus Linguistics in North America*, pages 191–205. Rodopi, 2004.

[4] M. Ghadessy, A. Henry, and R.L. Roseberry. *Preface. In Small Corpus Studies and ELT: Theory and Practice*. Studies in corpus linguistics. J. Benjamins Publishing Company, Amsterdam,Philadelphia, 2001.

[5] Rayid Ghani, Rosie Jones, and Dunja Mladenić. Mining the web to create minority language corpora. In *Proceedings of the Tenth International Conference on Information and Knowledge Management*, CIKM '01, pages 279–286, New York, NY, USA, 2001. ACM.

[6] Jiawei Han and Kevin Chang. Data mining for web intelligence. *Computer*, 35(11):64–70, November 2002.

[7] Andrew Kehoe and Renouf Antoinette. WebCorp: Applying the Web to Linguistics and Linguistics to the Web. In *WWW2002 Conference*, 2002.

[8] Adam Kilgarriff, Siva Reddy, Jan Pomiklek, and Avinesh PVS. A corpus factory for many languages. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may 2010. European Language Resources Association (ELRA).

[9] Christian Kohlschütter, Peter Fankhauser, and Wolfgang Nejdl. Boilerplate detection using shallow text features. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, pages 441–450, New York, NY, USA, 2010. ACM.

[10] Yi Lan, Liu Bing, and Li Xiaoli. Eliminating noisy information in web pages for data mining. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 296–305, New York, NY, USA, 2003. ACM.

[11] T. McEnery and A. Wilson. *Corpus linguistics*. Edinburgh textbooks in empirical linguistics. Edinburgh University Press, 1996.

[12] Otman Menad and Gilles Bernard. Mesure de la similitude entre blocs de donnes pour lextraction automatique du contenu web. *21me Rencontre de la Socit Francophone de Classication*, 2014.

[13] Georgios Petasis and Dimitrios Petasis. Blogbuster: A tool for extracting corpora from the blogosphere. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may 2010. European Language Resources Association (ELRA).

[14] Resnik Philip and Elkiss Aaron. The linguist's search engine: An overview. In *Proceedings of the ACL 2005 on Interactive Poster and Demonstration Sessions*, ACLdemo '05, pages 33–36, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[15] Jakub Piskorski and Roman Yangarber. Information extraction: Past, present and future. In Thierry Poibeau, Horacio Saggion, Jakub Piskorski, and Roman Yangarber, editors, *Multi-source, Multilingual Information Extraction and Summarization*, Theory and Applications of Natural Language Processing, chapter 2, pages 23–49. Springer Berlin Heidelberg, 2013.

[16] Serge Sharoff. Creating general-purpose corpora using automated search engine queries. In *WaCky! Working papers on the Web as Corpus. Gedit*, 2006.

[17] Lin Shian-Hua and Ho Jan-Ming. Discovering informative content blocks from web documents. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 588–593. ACM, 2002.