A Combined Approach to Dynamic Web Page Classification: Merging Structure and Content

Maria Niarou, Sofia Stamou
Department of Archives, Library and Museum Studies
Ionian University
Corfu, Greece
E-mails: {l14niar, stamou}@ionio.gr

Abstract — **Web data is constantly increasing at a very high pace. So does the need to come up with methods and tools that are able to process, organize and store this data effectively. To meet this need, several approaches have been proposed in the literature over the last decades, a critical amount of which focus on methods for classifying Web content in order to be able to retrieve relevant information in a cost-effective yet effortless manner. Motivated by the observation that the Web is changing not only with respect to content but also with respect to structure, we designed a combined classification method that encounters both textual and structural elements in the Web pages under examination. Our classification approach, presented here, investigates a number of parameters before assigning a Web page to a suitable category(-ies). A preliminary experimental evaluation of our method indicates that it accurately classifies web content both thematically and structurally.**

Keywords - Web data; dynamic data; similarities; semantics; classification; Web data structure.

I. INTRODUCTION

Many researchers have studied Web pages' classification. Existing research falls in two main categories: content-based and structure-based classification. Content-based classification tries to assign every Web page to a suitable thematic category [1]. To enable that, many approaches have been proposed for processing the textual content in Web pages, extracting thematic keywords, mapping them to existing ontologies and, therefore, identifying the most appropriate theme of the page. On the other hand, structure-based classification [2] relies on the pages' structural properties such as links, images etc. for grouping together pages of similar structure.

Despite the effectiveness of many of the proposed approaches, Web data classification still remains an open research challenge, basically because Web data is: (a) voluminous, (b) heterogeneous and (c) dynamic. In particular, the voluminous amount of online data makes it practically impossible to categorize every Web page regardless of the resources' power and availability. Web content is largely heterogeneous as it is represented via text, audio-visual material, multimedia, etc. Heterogeneity suggests that different elements should be encountered when trying to classify Web data and that different tools should be employed for elements' processing. This variance entails considerable communication overhead and increased complexity in any classification technique. Web content is dynamic and it constantly changes structurally and textually. Therefore, any classification attempt should account for the volatile nature of the material at hand and operate in a way that minimizes the need to re-examine already classified data unless it has significantly changed over time. The frequency of Web content change is also critical in Web data classification approaches because some pages might exhibit significant changes very often (e.g. news sites) while others might not change at all in their lifespan.

Driven by the idea that textual and structural classification are complementary, we designed a combined Web page classification approach that we present here. Our approach examines both content and structure for organizing Web pages and operates from an information retrieval point of view in the sense that it tries to group together pages that can serve similar information needs, thus lowering thus the cost and effort associated with user Web searches. Our method builds upon existing works and combines in a novel manner, elements in the Web pages' content and structure before concluding on the most appropriate category to assign every Web page. On top of that, our method accounts for the Web pages' changes (structural and textual) over time and, depending on the amount and frequency of changes, it reclassifies pages accordingly.

The experimental evaluation of our approach shows that our method manages to accurately classify Web pages when considering both their structure and contents, therefore implying that the combined investigation of structural and textual elements is successful in grouping together Web pages of similar themes or purposes.

The rest of the paper is organized as follows. In Section II, we review the related work. In Section III, we present the details of our classification approach and we justify the decisions we made with respect to the considered elements. Our approach combines three distinct yet complementary algorithms, which we discuss in detail. Section IV presents a preliminary evaluation we carried out in order to assess the effectiveness of our approach and reports the obtained results. We conclude the paper in Section V, where we also sketch our plans for future work.

II. RELATED WORK

Web data classification has been a research challenge over the last decades [3]. To this end, several approaches have been proposed in the literature aiming at the effective and automated classification of Web data. As already we mentioned in the previous section, the proposed methods fall into two main categories: those that rely on the analysis of Web pages' content in order to organize them into themati-

 ISBN: 978-1-61208-789-4

cally related groups, and those that exploit the pages' structural properties for enabling their classification.

The first approach focuses on the processing of text present in Web pages in order to learn the thematic categories it relates to. In this respect, existing works rely on machine learning for building classifiers, the most known of which are Naïve Bayes [4], Support Vector Machines [5], and decision trees [6]. Moreover, content-based classification techniques utilize semantic networks, ontologies, and hierarchies to create object clusters and, exploiting the relationships between the object categories, they organize the Web pages thematically [7]. The commonality across content-based classification methods is that they apply text pre-processing techniques to extract thematic keywords from text and, based on the frequency [8] and the (semantic) proximity of those keywords in the vector space representation, they are able to deduce their thematic category. Content-based classification can be greatly successful when classification algorithms have undergone a thorough training phase [9].

On the other hand, structure-based classification tries to organize Web pages based on their link properties [10], user clicks [11], sentiment analysis [12], etc. Despite the reduced time and effort associated with structural Web page classification, URL-based classifiers have to deal with a very small amount of information present in URLs, which is also noisy and may contain irrelevant features. Therefore, feature selection methods need to be applied, a process that increases the complexity of classification systems [13].

Although many of the above-mentioned techniques have proven successful, Web pages' classification still remains a challenging research quest for several reasons, but foremost because of the Web's volatile nature, which requires the re-examination of already classified Web pages. In this paper, we built upon existing research and we propose a classification approach that accounts for both the pages' content and structure in order to assign them to a suitable category. In addition, our method regularly re-examines classified data and, upon the detection of accountable changes, it re-classifies them accordingly.

Our approach differs from existing techniques as it is multi-dimensional and, at the same time, confronts the dynamic nature of the Web. Specifically, we propose a holistic approach for the Web pages' classification, exploiting a variety of features, some of which have been less used in existing work, and which, to the best of our knowledge, have not been combined into a single technique. At the same time, the methodology proposed integrates solutions that have previously been studied separately. Lastly, our classification method tries not only to identify a suitable category for assigning every Web page, but it also accounts for the changes a page might undergo, which in turn might require the re-classification of the page. The advantages of our proposed method are that it incorporates into a single technique the structural and content organization of Web pages. Moreover, our method regularly re-examines classified pages in order to detect changes and, upon doing so, to deduce if re-classification is needed. The details of our approach are presented next.

III. METHODOLOGY

Our classification approach is established on the ground that the content (mainly textual) of a Web page is informative of the subject(-s)/theme(-s) the page deals with, while the structure of the Web page hides information about its type, i.e., the intentions associated with user visits to its content. Motivated by the work of Jansen et al. [14], who showed that Web searches can be classified as either Navigational, Informational or Transactional and the significant impact this work had on Search Engine Optimization (SEO) approaches, we reverse the argument and we claim that Web pages can be classified in the same manner, i.e., as being either Informational, Transactional or Navigational. For instance, a Web page with a disproportionally small amount of text compared to non-textual elements - such as, links- is less likely to be an Informative page. Conversely, a page that requests the user to take some action on it (e.g., buy, pay, book) is more probably a Transactional page. Therefore, any attempt to classify Web content should account for both content and structure in the sense that content gives the theme and structure gives the type.

Taking all the above as our baseline assumption, we designed a classification algorithm that considers two complementary sets of Web page features: structural and textual. Our algorithm incorporates several components and operates on two complementary phases, namely structure-based and content-based classification, as described next.

A. MultiDimensional Page Classification

1) Structure-Based Classification

In Phase 1 (Page Type Recognition), the algorithm identifies the type of every page as follows. Given a page, it performs basic data processing in order to discriminate the page's textual and structural elements and then it utilizes a Text-to-Link-Analyzer [15] for extracting the page hyperlinks. Thereafter, it extracts the anchor text from every hyperlink and maps it against a list of transaction terms that are fed as input to the algorithm. The list of transaction terms has been manually constructed based on both empirical evidence and the findings of previous works [16]. Upon the detection of Transactional terms, it assigns a temporary tag to the page (i.e., type = *Transactional*) and further process it in Phase 2, as we will discuss next. In case no transactional terms are detected, the algorithm estimates the ratio between the page word-tokens to links and, if the ratio exceeds a given threshold t (experimentally set), it temporarily annotates the page as *Informational* and further process it in later steps. Finally, if the given page exhibits significantly more links than textual elements and lacks the presence of transactional terms, the algorithm annotates its type as *Navigational* and further process it in Phase 2. At the end of Phase 1, our classification algorithm enables a quick grouping of the pages under examination into one of the three categories, i.e., Informational, Navigational, Transactional, depending on their structural properties. To fine-tune this preliminary classification, we proceed to Phase 2 (Layered Page Classification) as follows.

The algorithm's input in Phase 2 consists of only the pages that have so far been characterized as either Transactional

or Navigational along with a table of transaction correspondences, T(corr), a table of terms signifying payment actions, T(payment), and a list of Web top-level domains, D(top) [17]. The Transaction correspondences table, T(corr), lists the Transactional terms under the compatible transactional category and it was built based on the findings of an earlier unpublished work, where we asked human experts to identify within a set of Web pages the terminology (mainly verbs) that indicates actions that need to be taken on Web pages on behalf of users. The results were cross-evaluated with Google AdWords [18] before ending them up to the final list. Alternatively, one could apply hidden Web crawling techniques for determining the most common transactional terms within Web sites, but this puts an extra burden on the classification process, which is out of scope for our work. Similarly, the table of terms signifying payment actions has been manually defined based on manual linguistic analysis by experts of available domain-specific vocabularies. Lastly, the list of top-level domains was determined based on [19]. In Appendix B, indicative examples of the aforementioned table contents are given.

Taking the above input, during Phase 2, the algorithm begins with the likely Transactional pages and maps their transactional terms against the T(corr) table. It then estimates the occurrence frequency of every mapping found and tags the page with the most frequently occurring term. This term implies the type of transaction (e.g., pay, book, register, play) performed on the page. As an additional step, the algorithm maps the transactional terms of the page against the T(payment) table and, if there is a matching found, it characterizes the page as "not-free", otherwise it characterizes it as "free". This step figuratively validates the transaction. At the end of this step, every page that has been preliminary identified as Transactional is verified against terminological resources and, upon such verification, it is annotated with the type of transaction it entails and it is then classified as Transactional.

Afterwards, the algorithm examines the pages that have been preliminary characterized as Navigational and, based on their URL properties (e.g., number of '/', URL suffix, number of contained URLs), it annotates them either as Navigational_Homepage, if their URLs map against the list of top-level Web domains (D(top)), or as Navigational_Web page if they exhibit '/' above a threshold h (experimentally determined) and/or if they contain valid internal links. At the end of this step, every page that has been preliminary identified as Navigational, is either annotated as Navigational_Homepage/ Navigational_Web page or sent back to Phase1 for reprocessing. Figure 1 shows the pseudo-code of Procedure 1 of the algorithm, which classifies the pages structurally.

Having completed these two phases, our algorithm classifies Web pages into the most appropriate type (Transactional, Informational, Navigational) depending predominantly on their structural properties. The next procedure is to further examine the pages that have been classified by type and detect the main theme discussed in their contents so as to enable thematic classification.

```
ALGORITHM1: Multi-Dimensional Page Classification
PROCEDURE1: Structure-Based Classification
Phase1: Page Type Recognition
Input: P, tokenizer, T(trans), Text-to-Link-Analyzer, (t)
    for every P
        look for t(trans) appearing as link
            if any
                tag P as P(transactional)
            else
                compute word tokens to links ratio (R)
                    if R≥ t
                        tag P as P(informational)
                    else
                        tag P as P(navigational)
                        end
Output: P(transactional), P(navigational), P(informational)
Phase2: Layered Page Classification given the Type
Input: P(transactional), P(navigational), T(corr), T(payment) D(top), LinkC, (h)
    for every P(transactional)
        map P(transactional) to the Table (corr)
            for every mapping found
                count occurrences and tag P(transactional) with the category of max occurrence
            else
                look for t(payment) appearing as link
                    if t(payment) ≥ 1
                        tag P(transactional) as "not-free"
                    else
                        tag P(transactional) as "free"
                        end
    for every P(navigational) starting after "http(s)://"
        count the number of "/" in url
            if "/" ≥ h
                tag P(navigational) as "WebPage" and
                set the number of "/" as depth value
                end
            else
                tag P(navigational) as "HomePage" and
                map the HomePage suffix to the D(top)
                    if there is a mapping
                        tag HomePage with the suffix meaning
                        end
                    else
                        validate url against LinkC
                            for every valid link
                                if internal
                                    set the number of (/) as depth value
                                    end
                                else
                                    send P(navigational) to Procedure1
                                    end
end
Output: Structure-Based layered classified pages
```

Figure 1. MultiDimensional Page Classification_Procedure1 (Structure-Based Classification)

2) *Textual Based Classification*

To enable thematic classification, the algorithm begins by extracting textual elements from the page, anchor title and title. Having experimented with several textual features, we ended up with anchor title and title as the most informative of the theme of a page. Extracted anchor title and title terms are cross-matched and, upon the detection of exactly matching terms among them, we use those terms to annotate the theme of the page. Unless exact matchings are found, we apply traditional keyword extraction techniques to the pages' body and based on the top n-appearing keywords, we map them to WordNet lexical hierarchy [20]. Upon the detection of keyword mapping synsets, we extract the glosses of the latter and we look for overlapping terms within their definitions, i.e., glosses. The definition terms that are frequently overlapping across keyword matching glosses are utilized for verbalizing the theme of the page. Unless there are matchings between page keywords and WordNet synsets or unless there are no overlapping definition terms in the glosses of matching keywords, the theme of the page is deemed 'unknown' and the page is left unclassified. In Figure 2, we illustrate the pseudo-code of Procedure 2 of the algorithm.

 ISBN: 978-1-61208-789-4

```
ALGORITHM1: Multi-Dimensional Page Classification
PROCEDURE2: Content-Based Classification
Phase1: Textual Elements Extraction
Input: P
    for each P
        search for anchor title in Url
            if any
                tag as "P's anchorTitle"
                end
            else
                search for title in text body
                    if any
                        tag as "P's textTitle"
                        end
end
Output: P tagged with Textual elements
Phase2: Theme Detection
Input: (P's anchorTitle), (P's textTitle), WebPage Word Counter, PoS-Tagger,
Parser, WordNet, lemmatizer, (TF*IDF), (n)
For every page P look for common terms between P's anchorTitle and P's
textTitle
    if found
        use common terms as the theme(-s) to tag P
        end
    else
        PoS-tag and lemmatize P's text and extract the first n-appearing
    keywords
        check for overlapping terms between P's keywords and (P's anchor
        title and P's text title)
            if found
                use overlapping terms as the theme(-s) to tag P
                end
            else
                map P's first n-appearing keywords to WordNet and
                look for common senses between P's keywords and
                (P's anchor title and P's text title)
                    if found
                        use terms of common senses as the
                    theme(-s) to tag P
                        end
                    else
                        tag P as of unknown category
                    (Punknown)
```

Figure 2. MultiDimensional Page Classification_Procedure 2 (Content-Based Classification)

Based on the above phases and procedures, our algorithm classifies Web pages both by content and type. The output of the algorithm is the input Web pages annotated with a label indicating the exact type of the page as well as the theme of the contained information. What should be stressed is that the algorithm operates on the assumption that every page on the Web has a predominant intention, i.e., type, and, as such, the algorithm tries to detect this type and proceed accordingly. If there are pages of mixed types (e.g., Transactional and Informational) and those types are equally pronounced in their content, the algorithm deems the page type 'unknown' and proceeds with the textual classification of the page. To enable Web page classification in multiple types, we should adjust the threshold values accordingly. We defer this study for future work.

B. ReClassification based on Change Detection

A common challenge in Web data classification methods is how to account for the Web pages' dynamic nature, i.e., how to ensure that the classification outcome is up-to-date. To account for that, researchers have proposed various techniques for detecting Web changes [21] [22]. Such changes might be encountered to the content and/or structure of existing pages or they might concern the death or the birth of new pages.

Given that the Web is constantly evolving, we incorporated a re-classification component to our algorithm. The goal here is to detect, measure and identify the changes that possibly occur in the Web pages already classified both structurally and thematically. This procedure highlights the Web pages that need to be re-classified, in order to maintain data indexes always updated.

```
ALGORITHM2: Re-Classification based on Change Detection
Input: P(class, T), P'(unclass, T')
Procedure1: Re-Classification Decision based on Textual Changes
Input: (E(t) ∈ P), (E(t) ∈ P'), smlrtMetric, (m), (z)
    for each pair of (Pi ∈ P(class,T), (P'i ∈ P'(unclass, T'))
        compute sim(Pi, P'i)
            if sim(Pi, P'i) ≥ m
                tag P'i as thematically unchanged and classify P'i to the category of Pi
                end
            else
                tag P'i as thematically changed and
                compare (E(t) ∈ P'i) with (E(t) ∈ Pi)
                    count ((E(t) ∈ P'i) ≠ (E(t) ∈ Pi))
                        if ((E(t) ∈ P'i) ≠ (E(t) ∈ Pi)) ≤ z
                        go to Algorithm2Procedure2
                        end
                    else
                        send P'i to Algorithm1Procedure2
                        end
end
Output: thematically unchanged pages P' over time T'
Procedure2: Re-Classification Decision based on Structural Changes
Input: (E(s) ∈ P), (E(s) ∈ P'), smlrtMetric, (z)
    for each pair of ((Pi ∈ P(class, T), (P'i ∈ P'(unclass, T'))
        compare (E(s) ∈ Pi) with (E(s) ∈ P'i) and
        count ((E(s) ∈ Pi) ≠ (E(s) ∈ P'i))
            if ((E(s) ∈ Pi) ≠ (E(s) ∈ P'i)) ≤ z
                tag P'i as structurally unchanged and classify P'i to the category of Pi
                end
            else
                send P'i to Algorithm1Procedure1
                end
Output: structurally unchanged pages P' over time T'
Output: P'(ReClass, T'), thematically unchanged pages P' over time T', structurally unchanged pages P' over
time T'
```

Figure 3. Re-Classification based on Change Detection

According to Algorithm 2 (pseudocode shown in Figure 3), after its initialization, it compares, via similarity metrics, the structural and textual elements of any given page with their counterparts previously identified during the page's initial classification. The similarity metrics used in our approach are the Tree Edit Distance measures and Jaccard coefficient [23]. Based on the above, if the similarity between the pages' elements falls behind a predefined threshold value, the algorithm considers the page as changed and sends it back to Algorithm 1 for textual and/or structural (re-)classification.

Conversely, if both the structural and the textual elements of the page remain the same over time, the algorithm considers the page as unchanged and retains it to the category(-ies)

it has been initially assigned by the classification algorithm. Note here that the value of thresholds can be experimentally fixed depending on the available data and the sought classification precision. Moreover, one could adjust thresholds dynamically, but, in the course of our experiment (as we discuss next), threshold values have been pre-determined.

C. *Optimized Re-Classification based on Change's Frequency Detection*

Having addressed the issue of changing Web content, we take a step further and we account for the changes' frequency. Change's frequency detection of a page, helps us determine our re-classification policy in order to save time and resources. In this framework, we capture the frequency with which Web pages change in order to optimize the runs of our Re-Classification algorithm. The idea was inspired by the work of Meegahapola et al. [24] and driven by the fact that Web pages change at different frequency rates.

According to the pseudocode of the algorithm, illustrated by Figure 4, we adjusted a timer, which is activated upon the initialization of the Re-Classification algorithm. The time intervals between the initialization of the Re-Classification runs are also predefined based on experimental set. Every time a change is detected between two chronologically different snapshots of a page, the timer records it. After several iterations of the Re-Classification algorithm on a single page at different time intervals, all the changes the page has undergone are recorded by the timer along with the timestamp of the change detection. This timeline of Web page changes helps us determine the best time period for a page to be revisited for change detection and if needed for re-classification.

```
Algorithm3: Optimized Re-Classification based on Change's Frequency Detection
Input: (P(class, T)), ((E(t) U E(s)) ∈ P(class, T)), P' ⊆ (P'(re-class, T'), ((E(t) U E(s)) ∈ P'(reClass,T')),
MaxFreqChange, MinFreqChange, Timer
    when Algorithm2 initializes, record Ts
    for every pair of ((Pi ∈ P(class, T), (P'i ∈ P'(re-class, T'))
        set Timer
            while ((E(t) U E(s)) ∈ Pi(class, T)) ≠ ((E(t) U E(s)) ∈ P'i(re-class,T')), record Ts
                if Ts ≥ MaxFreqChange
                    tag P'i as HighlyChanging Page and keep it in a secondary Index
                    end
                else
                    if Ts ≤ MinFreqChange
                        tag P'i as RarelyChanging Page and keep it in a secondary Index
                        end
                    else
                        tag P'i as RegularlyChanging Page and send it to Algorithm2
                        end
end
Output: Selection of Pages that need periodical Re-Classification
```

Figure 4. Optimized ReClassification based on Change's Frequency Detection.

Based on the above process, our method ensures that classified pages which undergo regular changes, are reconsidered by our classification algorithm when needed, in order to maintain their organization up-to-date.

Next, we present the experimental evaluation of our method and we report the obtained results.

IV. EXPERIMENTAL IMPLEMENTATION AND EVALUATION

To evaluate the effectiveness of our classification algorithm, we carried out a small-scale experiment in which we validated: (a) the classification performance of our method, and (b) the potential weaknesses of our approach. To collect our experimental dataset, we asked 10 experienced Web users to provide us with their bookmarked pages.

We informed our volunteers about our study objectives and we asked them to indicate for each of their shared bookmark the type of the page (by selecting between Informational, Transactional and Navigational) and the theme of the page. To familiarize our participants with the page types, we gave them brief instructions with respect to the definition of every type and we trained them by giving several examples. Moreover, we instructed them to indicate a single structural type for every page and in case of uncertainty to remove the page from their selection list. On the other hand, the theme of every page was self-determined by our volunteers and verbalized based on their understanding of the page's theme. We instructed our subjects to use as many keywords as they wished for verbalizing the underlying theme of a page, but upon the indication of several thematic keywords we asked them to point out the one that was in their opinion the predominant. The volunteers who supplied us with data were not further involved in the experimental process.

Based on the above dataset, we ended up with a gold-standard test-data of 2,330 pages, each of which was manually labeled by our participants with both structural type and thematic information. In TABLE I. we summarize the statistics of our experimental dataset.

TABLE I. EXPERIMENTAL TEST SET STATISTICS

Total set of experimental pages	***2,330***
Percentage of Informational pages	63%
Percentage of Transactional pages	17.99%
Percentage of Navigational pages	19.01%

This test set (cleaned up from any labelling) was given as input to our classification algorithm. Before proceeding with the details of our experiment, we should stress that the algorithm did not undergo any training phase, but rather it run in several iterations in order to fix the values of the thresholds it incorporates. During the first iteration, each threshold value was uniformly set to 0.5 suggesting equal probabilities so as to avoid any bias. Thereafter, in every subsequent iteration the values of each threshold were fine-tuned and based on the median values of all iterations, they were fixed as follows: the word tokens/links ratio (R) for discriminating between Informational and Navigational pages was set to 60/40 the number (h) of "/" in every page URL was set to 2, for the thematic classification we exploited the top 5 appearing keywords, whereas we retained the threshold values associated with the change detection algorithm to 0.5. Lastly, *MaxFreqChange* and *MinFreqChange* values

were set to 2 and 1 respectively. Of course, one could modify the applied thresholds depending on the experimental set, or she/he could fix their values uniformly or finally determine thresholds dynamically depending on the application domain of the algorithm. Further experimenting with alternative threshold values falls beyond the scope of this study.

Having collected and pre-processed our experimental test pages and having determined threshold values, we run our algorithm and we evaluated its classification accuracy as follows. We compared both the structural and thematic categories our algorithm identified for each of the experimental pages against the respective structural and thematic categories our participants had manually indicated for the corresponding pages. That is, we evaluated the algorithm's structure-based classification accuracy (i.e. the ability to discriminate Informational, Navigational or Transactional pages) by comparing the structural type tags our participants had manually indicated to the tags our algorithm identified for labelling the respective pages. The matching tags across pages were deemed as correct structural classifications (true positives), whereas mismatching tags flagged shortcomings in the algorithm.

Similarly, we evaluated the algorithm's content-based classification accuracy (i.e. the ability to identify a suitable theme for every page) by comparing the list of thematic keywords our participants had indicated for every page to the thematic terms our algorithm had automatically identified for each page. Matching thematic keywords across pages were deemed as correct content-based classifications (true positives), whereas mismatches were interpreted as the algorithm's failure to identify a suitable thematic category for a page. At this point, we should stress that the terminology used for naming a thematic category was not always identical between that supplied by our volunteers and the terminology identified by our algorithm. Thus, to enable the comparison between the two we relied on WordNet against which we estimated the semantic similarity between the two. For measuring similarity, we utilized the Wu and Palmer similarity metric [25]. If the similarity values between terms (automatically detected by the algorithm and manually defined by our subjects) exceeded the value of 0.8 (values range between 0=no similarity and 1= exact match) we deemed the theme the algorithm identified as correct (i.e. true positive) else we deemed the theme as wrong (i.e. false positive).

The metrics we used for quantifying the algorithm's accuracy are classification recall and precision. ***Classification recall*** estimates the proportion of pages that the algorithm classified correctly (*TP: true positives*) out of all the pages examined in the test-set (*TP+FN: true positives+false negatives*), whereas ***classification precision*** indicates the proportion of pages that the algorithm classified correctly (*TP: true positives*) out of all the pages the algorithm managed to classify (*TP+FP: true positives+false positives*). Classification recall shows the algorithm's capacity in identifying a category for every page it examines, whereas classification precision shows the algorithm's capacity in identifying the correct category of a Web page. The formulas of the two metrics are given below:

$$recall = \frac{true\ postives}{true\ posititves + false\ negatives}$$

$$precision = \frac{true\ positives}{true\ positives + false\ positives}$$

Out of the 2,330 experimental pages, our algorithm managed to classify 1,966 (84.3%) and the remaining 364 (15.7%) pages were assigned to the class '*unknown*', meaning that no category could be identified by the algorithm for those pages. Results suggest that the algorithm is successful in detecting a category (structural and thematic) for the majority of the examined pages and a manual inspection to the pages it left unclassified revealed that this was due to multiple structural nature of the no tagged pages and lack of text in the thematically untagged pages.

To evaluate the algorithm's success in correctly classifying Web pages we applied the precision and recall metrics, as previously explained, and we report obtained results in TABLE II. and Figure 5, respectively.

TABLE II. CLASSIFICATION RECALL AND PRECISION VALUES

Recall on Navigational pages	0.75
Precision on Navigational pages	1
Recall on Informational pages	0.89
Precision on Informational pages	0.98
Recall on Transactional pages	0.78
Precision on Transactional pages	1
Recall on thematic classification	0.83
Precision on thematic classification	0.87
Overall classification recall	**0.8**
Overall classification precision	**0.99**

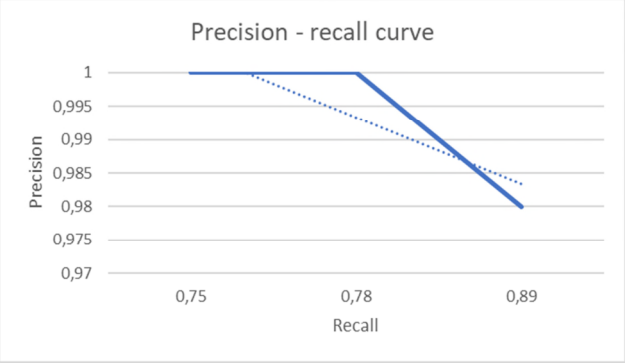

Figure 5. Classification precision-recall curve.

Results show that, out of all the Navigational pages in our test set (443 pages), the algorithm correctly identified 75% (332 pages) of them as Navigational and out of all the Informational pages in our test set (1,468 pages) the algorithm correctly identified 89% of them (1,307) as such. In addition, the algorithm correctly tagged as Transactional

78% (327 pages) of the examined Transactional pages (419 pages). Results show that the algorithm has a good overall accuracy in classifying pages based on their structural elements. This is further supported by the classification precision scores, which show that 100% of the pages the algorithm classified as Navigational were actually Navigational, 98% of the pages the algorithm classified as Informational were actually Informational and 100% of the pages the algorithm classified as Transactional were actually Transactional. Based on the classification precision and recall scores (overall and type-based), we may conclude that the algorithm has quite strict criteria for assigning Web pages to suitable structural types. This is attested by the very high precision scores and the lightly lower recall, which suggest that unless the algorithm has very strong indications of a page's type, it leaves it unclassified. A manual inspection of the cases the algorithm failed to identify the correct type of a page reveals that such pages had either mixed structural properties (e.g. their intention was both Informational and Navigational) or they contained very little information about their underlying type. With respect to the former, we already mentioned that in its current version the algorithm does not allow multiple structural classifications for a page, but this is an issue we are currently working on. To overcome the second shortcoming, we are testing additional elements that could signify the structural type of a page.

Regarding the algorithm's accuracy in identifying the theme of a page, our results show that the algorithm managed to identify a thematic category for 1,948 (i.e. 83.6%) of the 2,330 test pages and for those that it did find a category 87.4% (1,704 pages) were classified to the correct category (true positive classifications). Again, the manual inspection of our results showed that the algorithm failed to identify a theme for pages with very little content or with content verbalized with terms missing from WordNet. To overcome this terminological limitation, we are currently experimenting with a set of pre-defined categories to which we seek to organize the contextual elements of the Informational pages.

Thereafter, we evaluated the need for Web pages' reclassification as follows. We performed two additional downloads of the same set of pages after a period of one and three months respectively after the algorithm's first run. For every re-download, we run our re-classification algorithm as previously described and we computed the amount of changes between the content and structure elements of the examined pages, after one and three months since their first classification. The obtain results are reported in TABLE III. As the table shows, 67.9% of the pages had changed over a period of one month with the majority of changes being structural. In detail, 62% of the re-examined pages had undergone structural changes, 4.6% had undergone textual changes and only 1.3% of them had undergone both structural and textual charges. A close inspection to the amount of changes reveals that, the striking majority of them are minor (e.g. date changes) and thus there is no need to reclassify those pages. However, we found that in 6.6% of the changed pages, the structural and/or textual alterations were significant, therefore triggering the need to re-classify them.

Similarly, when re-examining the same set of pages after a period of three months, we found that 58.7% of them had changed, with the majority of changes pronounced mainly in structure (49.3%) and less in text (7.3%). Again, we observed that only a small portion of those pages (4.6%) exhibited significant alterations and, thus, should be reclassified.

As a last evaluation step, we computed for the pages that do need re-classification the frequency of changes they undergo over a period of three months since their first classification and we found that 46% of them change frequently, meaning that they had changed at least twice within a time slot of three months. This implies that for those changes regular re-examination is needed in order to keep classification up-to-date and that the algorithm should be periodically revisiting them. Based on the manual examination of a sample data out of those frequently changing pages, reveals that these concern among others news sites, online applications and so forth. Conversely, inspecting the results of the Re-Classification algorithm, when a page changes only with respect to the word tokens/links ratio (R), it is not being sent to Algorithm 1 for re-classification, as it is appearing with a small change percentage. However, this element is determined for page's type, so sometimes this change may be more significant than it seems. This downside could be overcome by transforming any structural and textual element to weighted element, according to its role in the classification decision. However, we defer this last issue for a follow up study of the current work.

TABLE III. CHANGING PAGES THAT NEED RECLASSIFICATION

Pages changed after 1 month	**67.9%**
Pages structurally changed	62%
Pages textually changed	4.6%
Pages structurally and textually changed	1.3%
Pages re-classified after 1 month	**6.6%**
Pages changed after 3 months	**58.7%**
Pages structurally changed	49.3%
Pages textually changed	7.3%
Pages structurally and textually changed	2.1%
Pages re-classified after 3 months	**4.6%**
Highly changing pages after 3 months	46%

V. CONCLUSION AND FUTURE WORK

In this paper, we present a novel Web pages classification approach that combines the structural properties and textual elements of Web pages in order to classify them in two dimensions, namely type and theme. Moreover, our method accounts for the changing nature of the Web and foresees a number of actions in order to determine whether a page needs to be re-classified after a time period as well as what is the best time period for re-classification. A prelimi-

nary experimental evaluation to our method against a manually annotated set of pages reveals that it manages to effectively capture both the type and the general theme of the examined Web pages.

We are currently improving our method to tackle some minor shortcomings already detected during experiments. More specifically, we are improving some parts of the Algorithm 1, so as to enable multiple type classifications of a given Web page if necessary. We are also experimenting with alternative/additional lexical resources for the detection of the pages' theme and we work on testing alternative similarity metrics. The prospective flexibility that characterizes our approach, respecting the resources mentioned above, would extend our methodology's significance. Close to the above, for the Transactional pages' detection, taking into consideration the images and/or symbols that represent a Transactional option for the user, and not only the verbal T(terms) appearing as links in the page's text body, can broaden algorithm's performance.

Observing the experimental results of the Re-Classification Algorithm, we grasped that the change of any textual and structural element is not equally important. In other words, each element can be less or more determinative for the page's re-classification decision. Based on this notice, we examine the elements assignment with weights. Additionally, in Algorithm 3, we consider as a challenge the dynamic definition of time intervals between the initialization of the Re-Classification runs. Additionally, we think about checking the algorithms' complexity and response time. Lastly, we are in the process of a large-scale experiment (larger dataset checked for longer time and after more time intervals), the results of which will be resented in a future work.

REFERENCES

[1] L. Safae, B. E. Habib, and T. Abderrahim, "A Review of Machine Learning Algorithms for Web Page Classification," in *2018 IEEE 5th International Congress on Information Science and Technology (CiSt)*, Oct. 2018, pp. 220–226, doi: 10.1109/CIST.2018.8596420.

[2] A. Kumar and R. K. Singh, "A Study on Web Structure Mining," vol. 04, no. 1, p. 6.

[3] C.-W. Tsai, C.-F. Lai, H.-C. Chao, and A. V. Vasilakos, "Big data analytics: a survey," *J. Big Data*, vol. 2, no. 1, p. 21, Oct. 2015, doi: 10.1186/s40537-015-0030-3.

[4] R. Rajalakshmi and C. Aravindan, "Naive Bayes Approach for Website Classification," in *Information Technology and Mobile Communication*, vol. 147, V. V. Das, G. Thomas, and F. Lumban Gaol, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 323–326.

[5] S. Shinde, P. Joeg, and S. Vanjale, "Web Document Classification using Support Vector Machine," in *2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC)*, Sep. 2017, pp. 688–691, doi: 10.1109/CTCEEC.2017.8455102.

[6] P. Kenekayoro, K. Buckley, and M. Thelwall, "Automatic classification of academic web page types," *Scientometrics*, vol. 101, no. 2, pp. 1015–1026, Nov. 2014, doi: 10.1007/s11192-014-1292-9.

[7] H. Li, Z. Xu, T. Li, G. Sun, and K.-K. Raymond Choo, "An optimized approach for massive web page classification using entity similarity based on semantic network," *Future Gener. Comput. Syst.*, vol. 76, pp. 510–518, Nov. 2017, doi: 10.1016/j.future.2017.03.003.

[8] A. Onan, S. Korukoğlu, and H. Bulut, "Ensemble of keyword extraction methods and classifiers in text classification," *Expert Syst. Appl.*, vol. 57, pp. 232–247, Sep. 2016, doi: 10.1016/j.eswa.2016.03.045.

[9] J. Qiu, Q. Wu, G. Ding, Y. Xu, and S. Feng, "A survey of machine learning for big data processing," *EURASIP J. Adv. Signal Process.*, vol. 2016, no. 1, p. 67, Dec. 2016, doi: 10.1186/s13634-016-0355-x.

[10] M. J. H. Mughal, "Data Mining: Web Data Mining Techniques, Tools and Algorithms: An Overview," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 6, 2018, doi: 10.14569/IJACSA.2018.090630.

[11] Y. Liu, J.-Y. Nie, and Y. Chang, "Constructing click models for search users," *Inf. Retr. J.*, vol. 20, no. 1, pp. 1–3, Feb. 2017, doi: 10.1007/s10791-017-9294-x.

[12] F. Hemmatian and M. K. Sohrabi, "A survey on classification-techniques for opinion mining and sentiment analysis," *Artif. Intell. Rev.*, vol. 52, no. 3, pp. 1495–1545, Oct. 2019, doi: 10.1007/s10462-017-9599-6.

[13] R. Rajalakshmi and C. Aravindan, "A Naive Bayes approach for URL classification with supervised feature selection and rejection framework," *Comput. Intell.*, vol. 34, no. 1, pp. 363–396, 2018, doi: 10.1111/coin.12158.

[14] B. J. Jansen, D. L. Booth, and A. Spink, "Determining the informational, navigational, and transactional intent of Web queries," *Inf. Process. Manag.*, vol. 44, no. 3, pp. 1251–1266, May 2008, doi: 10.1016/j.ipm.2007.07.015.

[15] *https://sonovabitc.win/analyze.php* [retrieved: May, 2020].

[16] B. J. Jansen, D. L. Booth, and A. Spink, "Determining the user intent of web search engine queries," in *Proceedings of the 16th international conference on World Wide Web - WWW '07*, Banff, Alberta, Canada, 2007, p. 1149, doi: 10.1145/1242572.1242739.

[17] *https://en.wikipedia.org/wiki/List_of_Internet_top-level_domains [retrieved: May, 2020]*.

[18] M. Raffinot and R. Rivière, "Optimizing Google Shopping Campaigns Structures with Query-Level Matching," *ArXiv170804586 Cs*, Aug. 2017, Accessed: Jul. 17, 2020. [Online]. Available: http://arxiv.org/abs/1708.04586.

[19] S. C. Jeeva and E. B. Rajsingh, "Intelligent phishing url detection using association rule mining," *Hum. -Centric Comput. Inf. Sci.*, vol. 6, no. 1, p. 10, Jul. 2016, doi: 10.1186/s13673-016-0064-3.

[20] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller, "Introduction to WordNet: An On-line Lexical Database," *Int. J. Lexicogr.*, vol. 3, no. 4, pp. 235–244, 1990, doi: 10.1093/ijl/3.4.235.

[21] S. Flesca and E. Masciari, "Efficient and effective Web change detection," *Data Knowl. Eng.*, vol. 46, no. 2, pp. 203–224, Aug. 2003, doi: 10.1016/S0169-023X (02)00210-0.

[22] D. Yadav, A. K. Sharma, and J. P. Gupta, "Change Detection in Web Pages," in *10th International Conference on Information Technology (ICIT 2007)*, Rourkela, Orissa, India, Dec. 2007, pp. 265–270, doi: 10.1109/ICIT.2007.37.

 ISBN: 978-1-61208-789-4

[23] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to data mining*, First edition, Pearson new international edition. Harlow: Pearson, 2014.

[24] L. Meegahapola, R. Alwis, E. Nimalarathna, V. Malawaarachchi, D. Meedeniya, and S. Jayarathna, "Detection of change frequency in web pages to optimize server-based scheduling," in *2017 Seventeenth International Conference on Advances in ICT for Emerging Regions (ICTer)*, Sep. 2017, pp. 1–7, doi: 10.1109/ICTER.2017.8257791.

[25] Z. Wu, M. Palmer, "Verb Semantics and Lexical Selection," pp. 133-138, 1994.

APPENDIX A

(E(s) ∈ P): list of P structural elements
(E(s) ∈ P'): list of P' structural elements
(E(t) ∈ P): list of P textual elements
(E(t) ∈ P'): list of P' textual elements
(E(t) U E(s)) ∈ P(class, T): list of P(class, T) textual and structural elements
(E(t) U E(s)) ∈ P'(re-class,T'): textual and structural elements of P'(re-class,T')
(h): threshold for homepages' detection
(z): threshold for structurally unchanged pages' detection
(n): threshold for keywords
(m): threshold for the thematically unchanged pages' detection
(t): threshold for the informational pages' detection
(TF*IDF): short for "term frequency–inverse document frequency", is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.
(Ts): time stamp
D(top): list of web top-level domains
Lemmatizer: tool that groups inflected forms together as a single base form
LinkC: link counter (tool)
MaxFreqChange: maximum frequency "allowed" by the algorithm for changes to webpages
MinFreqChange: minimum frequency "allowed" by the algorithm for changes to webpages
P' ⊆ (P'(re-class, T'): subset of P' that have been ReClassified based on Algorithm2 at time T'
P(class, T): pages classified from Algorithm1 at time T
P'(class, T'): pages classified from Algorithm1 at time T'
P(navigational): navigational pages
P(transactional): transactional pages
P(informational): informational pages
P: pages for classification
P'(ReClass, T'): textually or/and structurally changed pages P' over time that need to be ReClassified
(P's anchorTitle): page's anchor title
(P's textTitle): page's text title
Parser: compiler or interpreter component that breaks data into smaller elements for easy translation into another language.
PoS-Tagger: software that reads text in some language and assigns parts of speech to each word (and other token), such as noun, verb, adjective, etc.
smlrtMetric: similarity metric
T(corr): table with transactions' correspondences
T(payment): table with Payment terms [t(payment)]
t(payment): payment terms
T(trans): table with transactional terms
t(trans): transactional terms
Text-to-Link-Analyzer: tool for the calculation of WordTokens2Links Ratio
Timer: timer that calculates the time based on defined formula (1). If t is the time when a webpage is first examined, the timer will calculate every t_i time instance that we want to re-examine the same page, according to the formula (1): Each time instance of re-examination is derived from the multiplication product of its preceding time instance with the constant defined.

$$t_i=(t_i\text{-}1) * \text{constant} \quad (1)$$

Tokenizer: tool for the tokenization of the text
WebPage Word Counter: tool for keywords' extraction
WordNet: hierarchically organized dictionary

APPENDIX B

TRANSACTION CORRESPODENCES TABLE T(CORR).

Booking	Download	E-commerce	Entertainment	Software
Book a table	Download	Shop Now	Download	Download
Book Now	Free trial	(Add to) Bag	Find a table	Free trial
Find a table	Games	(Add to) Basket	Play	Games

TABLE WITH PAYMENT TERMS T(PAYMENT).

Payment Terms
Book Now
Buy Online
Buy
Pruduct+Price
Shop Now
Wish List

TABLE WITH SOME WEB TOP-LEVEL DOMAINS, D(TOP).

Name	Entity
.com	commercial
.org	organization
.net	network
.int	international organizations
.edu	education
.pr	Puerto Rico (United States)
.ps	Palestine
.pt	Portugal
.py	Paraguay