IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 4, 2020, accepted December 28, 2020, date of publication January 8, 2021, date of current version January 14, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3049793

Corpulyzer: A Novel Framework for Building Low Resource Language Corpora

BILAL TAHIR AND MUHAMMAD AMIR MEHMOOD
Al-Khawarizmi Institute of Computer Science, University of Engineering and Technology, Lahore 54890, Pakistan
Corresponding author: Bilal Tahir (bilal.tahir@kics.edu.pk)

This work was supported in part by the Higher Education Commission (HEC), Pakistan, and in part by the Ministry of Planning Development and Reforms through the National Center in Big Data and Cloud Computing.

ABSTRACT The rapid proliferation of artificial intelligence has led to the development of sophisticated cutting-edge systems in natural language processing and computational linguistics domains. These systems heavily rely on high-quality dataset/corpora for the training of deep-learning algorithms to develop precise models. The preparation of a high-quality gold standard corpus for natural language processing on a large scale is a challenging task due to the need of huge computational resources, accurate language identification models, and precise content parsing tools. This task is further exacerbated in case of regional languages due to the scarcity of web content. In this article, we propose a generic framework of Corpus Analyzer – *Corpulyzer* – a novel framework for building low resource language corpora. Our framework consists of corpus generation and corpus analyzer module. We demonstrate the efficacy of our framework by creating a high-quality large scale corpus for the Urdu language as a case study. Leveraging dataset from Common Crawl Corpus (CCC), first, we prepare a list of seed URLs by filtering the Urdu language webpages. Next, we use Corpulyzer to crawl the World-Wide-Web (WWW) over a period of four years (2016–2020). We build Urdu web corpus "UrduWeb20" that consists of 8.0 million Urdu webpages crawled from 6,590 websites. In addition, we propose Low-Resource Language (LRL) website scoring algorithm and *content-size filter* for language-focused crawling to achieve optimal use of computational resources. Moreover, we analyze UrduWeb20 using variety of traditional metrics such as web-traffic-rank, URL depth, duplicate documents, and vocabulary distribution along with our newly defined content-richness metrics. Furthermore, we compare different characteristics of our corpus with three datasets of CCC. In general, we observe that contrary to CCC that focuses on crawling the limited number of webpages from highly ranked Urdu websites, Corpulyzer performs an in-depth crawling of Urdu content-rich websites. Finally, we made available Corpulyzer framework for the research community for corpus building.

INDEX TERMS Common crawl, web crawling, text corpus, corpus analysis, regional languages corpora.

I. INTRODUCTION

Over the last decade, Artificial Intelligence (AI) has revolutionized the Natural Language Processing (NLP) and Computational Linguistics fields. Modern state-of-the-art NLP technologies are assisting humans to interact with machines using different modalities like text, speech, and vision [1]–[3]. For instance, sentiment/emotion analysis, plagiarism detection, intelligent assistants, chatbot, question answering systems, search engines, and recommender systems are now commercially successful AI-based systems [4]–[19]. Last year, only chatbots assist 1.4 billion users while generating the revenue of $2.6 billion [20]. The market based on Natural Language Generation (NLG) services like automated journalism, text summarization, and automatic business analytics are expected to grow to $825 million by 2023 with Compound Annual Growth Rate (CARG) of 20.3% [21]. The success of these AI-based systems is attributed to the advancement in deep learning models, superior high-performance computing, and availability of the large scale high-quality text corpora. Text corpora is a key enabler of different widely adopted AI-based systems such as

The associate editor coordinating the review of this manuscript and approving it for publication was Liviu-Adrian Cotfas.

machine translation system [22]; speech-to-text (STT) [23]; automatic information extraction and understanding systems [24]; chatbots and virtual-assistants [25].

Current AI-based systems are mainly designed for the English and European languages. A key reason for the development of such technologies is the availability of high-quality corpora of these languages [26]. On the other hand, the progress of similar systems for other languages is stagnant due to the scarcity of high-quality large scale text corpora. Asian languages such as Hindi, Bengali, Indonesian, Urdu, Marathi, and Turkish are widely spoken languages with 85-637 million speakers [27]. However, these languages fall into low-resource and low-density language categories due to the unavailability of gold standard text corpora. According to the survey, 20% (1.26 billion) population of the world speak the English language while only 4.8% (369 million) speakers use English as their first language [28]. With 80% of the world population unfamiliar to the English language, NLP systems based on deep learning for Low-Resource Languages (LRL) are imperative need of current times.

In general, World-Wide-Web (WWW) is used as a key source to develop high-quality text corpus for different natural languages [29]–[31]. However, crawling the whole WWW is a challenging task and requires huge computation, storage, and human resources. In addition, it requires explicit language identification models and content parsing tools. By crawling the WWW, large scale text corpora of English [32]–[34], Chinese [35], German [36], Arabic [37], and multi-lingual [38] content are build. The augmentation of these datasets is possible because there is sufficient web content produced in the target languages. Indeed, due to the paucity of online content in case of LRL, the data collection task is like finding a needle in the haystack. Common Crawl Corpus (CCC) [39] is an organization that maintains an open repository of web crawled data and provides regular crawls of WWW on monthly basis. However, building a corpus of a low-resource language from CCC is a challenging task due to: i) sampling techniques, ii) filtering of webpages of target languages, and iii) full parsing of CCC.

On the other hand, several key challenges need to be addressed if one decides to build a corpus of a low-resource language using indigenous resources. First, there is a need for a sufficient hardware infrastructure that consists of servers, storage, and networking devices. Since the majority of the online content consists of English and European languages, therefore, crawlers need to filter out such content in majority of the cases resulting in low yield rate and waste of computational resources. Second, the vast amount of webpages in low-resource languages also contain multi-lingual content. For instance, only 2.48-12.83% webpages of Asian languages have content in one language [40], [41]. The inclusion of multi-lingual content will result in low-quality corpora of a target language. In this article, we argue that in order to build high-quality corpora of low-resource languages, we need an intelligent framework. For this purpose, we propose ''Corpulyzer'' – a novel framework for building low resource language corpora. Our framework contains two modules: i) corpus generation and ii) corpus analyzer. The first module utilizes our proposed *LRL website scoring* algorithm and content filters to collect language-specific webpages with optimal usage of computational and storage resources. In particular, this module uses our Web-ArticleMiner (Web-AM) algorithm and *content-size filter* to mitigate content noise. The second module examines the developed corpus with respect to state-of-the-art metrics used in web measurements, Natural Language Processing (NLP), and Information Retrieval (IR). In literature, analysis of different developed corpora is generally performed by using characteristics of crawled content only. In general, language community has ignored key distinguishing characteristics of the crawled websites. We argue that an in-depth analysis of content sources is necessary to ascertain the quality of corpus. Therefore, our framework examines the crawled websites by performing an extensive analysis with variety of metrics like Web-traffic-rank, URL depth, and Churn rate. Moreover, two metrics of *webpage-language-share* and *website–language-richness* are introduced to examine the content-richness of LRL corpora.

In this work, we build upon our previous approach [40] where we developed a dataset consisting of 1.28 million Urdu webpages from CCC 2016 dataset. Our analysis on the dataset manifests the presence of 84% noisy webpages that motivated us to design a framework to build high-quality LRL corpora. We demonstrate the efficacy of our framework by preparing the high-quality Urdu Web Corpus (**UrduWeb20**) as the case study. First, we prepare a list of seed URLs by filtering Urdu webpages from CCC by using open-source language identification module of Compact-Language-Detector-2 (CLD2) [42]. With Corpulyzer framework, we crawl the WWW over a period of four years (2016-2020) using our own crawling infrastructure. Our corpus contains 8.0 million Urdu webpages crawled from 6,590 websites. Our analysis of the vocabulary of UrduWeb20 reveals the presence of 89.75% Urdu and 10.25% non-Urdu tokens. Moreover, we develop three datasets of *CC-Urdu-meta*, *CC-Urdu-html*, and *CC-Urdu-crawl* after filtering Urdu webpages from CCC for comparison. Our comparison of different datasets illustrates major CCC limitations for building LRL corpora. To the best of our knowledge, this is the first attempt to design a framework to prepare and analyze the high-quality large scale text corpus of low-resource languages. Our survey on existing Urdu language corpora indicates that UrduWeb20 is the most representative dataset of Urdu content crawled from WWW to date. We have released Corpulyzer[1] for the research community.

Our major contributions are as follows:

- We present a novel framework of ''Corpulyzer'' to prepare and analyze the text corpus of different low-resource natural languages.

[1] https://sourceforge.net/projects/corpulyzer-urdu/

- We propose *LRL website scoring* algorithm to increase the yield rate of our crawler. Our algorithm configures the crawler to prioritize the content-rich websites of the target language.
- With Corpulyzer, we prepare Urdu Web corpus (UrduWeb20) that consists of 8.0 million Urdu webpages crawled from 6,590 websites. Our UrduWeb20 is composed of rich vocabulary of 4.1 billion total and 13.1 million unique uni-gram tokens.
- We empirically calculate the threshold value of *bytes* in a webpage to design our *content-size filter*. Our calculation shows 256 bytes of content as an optimal value of the threshold for the selection of Urdu language webpage.
- We compare UrduWeb20 with three datasets of CCC. Our comparison indicates that for low-resource natural languages CCC is not a reliable resource due to biasness of CCC towards high-rank websites and filtering of webpages greater than 1MB.
- Our analysis on crawled websites indicates that well-known website ranking algorithms like Alexa traffic rank and Harmonic centrality are not suitable for the selection of websites to create LRL corpora.

The rest of our paper is structured as follows: Section II presents the related work and Section III introduces the definitions and formulas of terms used in our article. In Section IV, we present the Corpulyzer framework. Details of our case study to build the Urdu language corpus are provided in Section V. The detailed analysis of our developed Urdu language corpus is reported in Section VI. We compare *UrduWeb20* with other Urdu datasets in Section VII along with some salient NLP/IR applications. Finally, we conclude our paper in Section VIII.

II. RELATED WORK

With the rise of Internet, NLP, and IR research communities have explored various aspects of the online content. This include large scale corpus building [37], [43], [44], text corpus driven service development [45]–[47], online content distribution analysis [48]–[52], corpus characteristics investigation [53], [54], websites analysis [55]–[59], and webpages classification [60]–[63].

A. LRL CORPORA – WEB

A number of studies have focused on developing the corpus of different languages by filtering webpages from Common Crawl Corpus [64]–[67]. For instance, Veisi *et al.* [68] processed the content of CCC, published books, and magazines to develop the first Central Kurdish language text corpus – AsoSoft. It contains 0.458 million Kurdish documents composed of 188 million total and 4.66 million unique tokens. In another study, a framework of CCNet was developed to extract mono-lingual content from CCC [52]. CCNet utilizes FastText word-embeddings to calculate the distance between Common Crawl and Wikipedia webpages to identify the target language in the content. A similar framework of LanguageCrawl [69] filtered out language-specific webpages from CCC using CLD2 to develop Word2Vec language models of various languages. In addition, Dunn [51] developed multi-lingual corpora of 148 languages and 423 billion tokens from CCC. Despite the need for high compute and storage, few efforts have been made to crawl the World-Wide-Web (WWW) to develop text corpora of low-resource languages. Suwaileh *et al.* [37] crawled mono-lingual corpora of ArabicWeb16 containing 150.9 million webpages from 768K websites. In this study, authors customized their crawler to prioritize Arabic websites for crawling. Similarly, C4Corpus containing 12 million webpages from 53 languages is built by crawling the web [38]. For the development of C4Corpus, URLs from CCC were used as a seed and link graph methodology is utilized to fetch webpages. Similarly, Krasselt *et al.* [70] performed focused crawling to fetch Swiss content from news, governmental, parliamentary records, companies, and NGO websites to develop Swiss-AL corpus. Swiss-AL corpus contains 8 million texts and 1.55 billion tokens. Similarly, we built a Urdu language corpus of 1.28 million Urdu webpages from CC corpus of 2.87 billion webpages [40], [50].

B. LRL CORPORA – SOCIAL MEDIA

Today, social media is a rich source to develop text corpora for different NLP tools [71]–[75]. Leveraging the content of these social media platforms, Cross-Lingual Arabic Blog Alerts (COLABA) [76] project has focused on collecting Arabic content from different social media platforms like blogs, discussion forums, and chats to develop NLP tools. COLABA used the collected data to develop Dialectal Arabic Information Retrieval Assistant (DIRA) [77], a term expansion tool to generate dialect search terms with relevant morphological variations from English or standard Arabic query terms. Similarly, Ljubešic *et al.* [78] presented an open-source tool of TweetCat to develop large scale tweets corpora for small languages. In this study, the authors demonstrated the effectiveness of their tool by developing a corpus of Serbian and Slovene languages containing 26.0 and 4.5 million tweets, respectively. In addition, 300 million German language tweets were utilized to train word-embedding models [79].

C. CORPUS ANALYSIS

Finding the distribution of online content of different natural languages is a challenging task. Grefenstette and Nioche [48] conducted the first study to estimate the share of different languages in online content by leveraging open-source language detection library [80]. Their analysis of online webpages crawled from 1996 to 2000 revealed that the majority of webpages contain content in English, French, Deutsch, Russian, and Spanish language. Tan *et al.* [81] examined the dissimilarity of words in English corpora developed from social media and web content. Authors compared the word-embeddings of both corpora and their analysis indicates

TABLE 1. Background information and terminologies.

SR#	Term	Research area	Definition
1	Webpage	Web measurements	Webpage or page is document available online which can be accessed on Internet via unique assigned URL.
2	Website	Web measurements	Website is collection of publicly available webpages shared on a single domain name such as bbc.com.
3	URL depth	Web measurements	URL depth shows how deeply the current webpage is located in website hierarchy. It shows the difference of current webpage from the homepage of the website.
4	Jaccard similarity	Web measurements	Jaccard similarity calculates the ratio of distinct overlapping samples between two datasets.
5	Churn rate	Web measurements	Churn rate calculates the fraction of sample that varies between two datasets.
6	Web-traffic-rank	Web measurements	Traffic rank of a certain website is a relative score assigned after ranking all websites on the Internet with respect to amount of organic traffic. In this article, websites rank values from most popular traffic rank service of Alexa are used.
7	Vocabulary	Natural Language Processing	Vocabulary consists of unique-tokens which represents the distinct words in the corpora. Also, total-tokens highlight the total number of words irrespective of how often they are repeated.
			We refer language-tokens as words in corpus belonging to target language. Using the uni-code range of target language, we label words as non-language-tokens even they contain single character out of defined range.
8	Zipf's Law	Natural Language Processing	Zipf's law states that if we assign ranks to all words of language according to their frequencies in some long text, then the resulting frequency-rank distribution follows a very simple empirical law and plot of log (rank) vs log (frequency) will produce a straight line with slope -1.
9	Duplicates	Information Retrieval	Exact-duplicate webpages contain exactly same textual content.
			Near-duplicate webpages contain significantly similar content which varies by few words or sentence.

the usage of jargon, slang, and other informal words in social media corpus.

From the above literature survey, we observe that systems are developed to collect low-resource languages content by either filtering language-specific content from large scale repositories or by crawling the whole WWW. However, the quality of crawled corpora is investigated using limited metrics such as vocabulary distribution and document length. Leveraging these research efforts, we propose a Corpulyzer framework to develop large scale repositories for low-resource languages by crawling the WWW. Furthermore, Corpulyzer lays the foundation of extensive analysis of textual corpora by investigating various characteristics like web measurements, NLP/IR analysis, and content-richness.

III. BACKGROUND AND TERMINOLOGIES

In this section, we discuss important background information and terminologies used in web measurements, natural language processing, and information retrieval research related to natural language corpora.

A. WEB MEASUREMENTS

A *webpage* or a web document is a collection of information available online that can be displayed using a browser via a unique URL address. The collection of these webpages sharing the same domain name is called as *website*. In addition, the URL hierarchy of webpages is used to investigate the extent to which crawler is fetching content from each website. A high *URL depth* – calculated after splitting a URL using delimiter of '/' [82] – indicates that crawler is able to crawl more content from a particular website. Moreover, *Jaccard similarity* of URLs in different corpora is calculated to find the overlapping webpages. Jaccard similarity calculates the ratio of distinct overlapping samples between two datasets. For example, Jaccard similarity coefficients ($sim_{Jaccard}$) between two datasets A and B that contain URLs of webpages is calculated using the Equation 1:

$$sim_{Jaccard} = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

Similarly, *churn-rate* is a key metric used to analyze the overlap of websites between two datasets, e.g., if the first dataset contains 100 websites and the second dataset contains 70 websites from the first dataset, then the churn rate is 30%. Finally, *web-traffic-rank* of a website is a relative score assigned after ranking all websites on the Internet with respect to the amount of their organic web traffic. The web-traffic-rank of a website shows trust of users on the information and quality of webpages hosted by that website. These terms are briefly summarized in Table 1.

B. NATURAL LANGUAGE PROCESSING (NLP)

The analysis of different aspects of the vocabulary of a language is a key component for corpora preparation. *Unique-tokens* represent distinct words in corpora while *total-tokens* highlight the total number of words irrespective of how often they are repeated. In addition, *language-tokens* are referred to words in the corpus belonging to a target language after filtering *non-language-tokens* using uni-codes of the target language. In natural language processing, Zipf's law is often used to study the relationship between word-frequency and the word-rank. Zipf's law states that if we assign *ranks* to all words of language according to their *frequencie*s in some

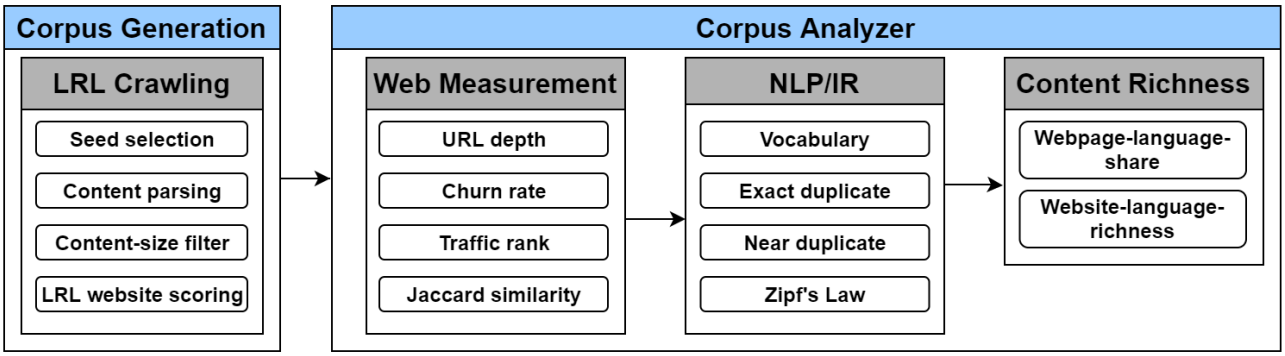


FIGURE 1. Architecture of Corpulyzer framework.

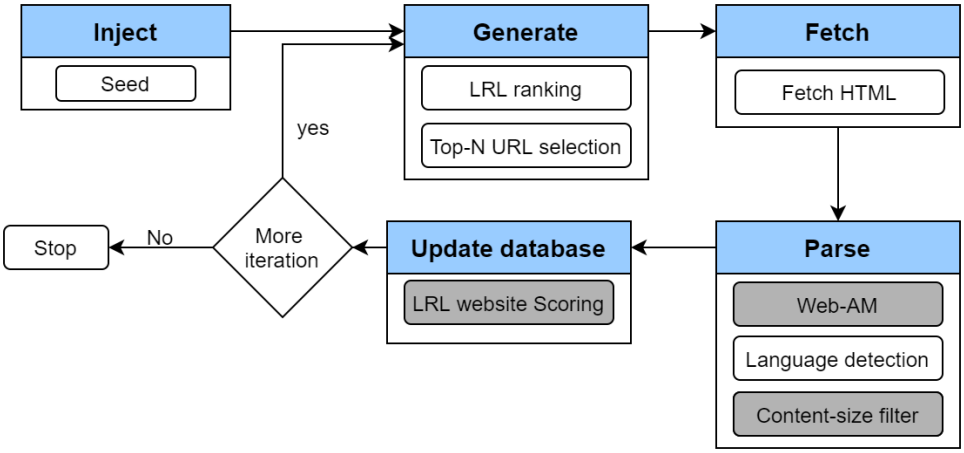


FIGURE 2. Web crawler – Flow diagram of different phases.

long text, then the resulting frequency-rank distribution follows a very simple empirical law and plot of log (rank) vs log (frequency) will produce a straight line with slope -1.

C. INFORMATION RETRIEVAL (IR)

The diversity of corpus is a crucial characteristic to judge the quality of an IR system. A key metric to examine the diversity is to find the duplicate webpages in a corpus. In general, there are two classes of duplicate webpages, i.e., *exact-duplicates* and *near-duplicates*. Webpages with same textual content are called *exact-duplicates*. For example, two webpages with content "I am a reader" are exact-duplicates of each other. On the other hand, webpages with similar but not same content are known as *near-duplicates*. The content of these webpages may differ by few words or sentences. For instance, two webpages with content of "i am a reader" and "i am a slow reader" are near-duplicates.

IV. CORPULYZER FRAMEWORK

In this section, we describe the architecture of the Corpulyzer framework. First, we present different components of the corpus generation module. Next, we discuss the evaluation metrics proposed for corpus analysis to gain insights into the crawled content.

A. CORPUS GENERATION

Figure 1 shows the Corpulyzer framework that consists of two major modules: corpus generation and corpus analyzer. The corpus generation module requires four essential components: i) LRL crawling, ii) content parsing, iii) content filtering, and iv) LRL website scoring. Next, we dive into the details of these components.

1) LRL CRAWLING

Web crawling is a vexing problem and is considered as the heart of any corpus generation process. The purpose of the web crawler is to discover, collect, and index webpages available online. In particular, a web crawler consists of five phases of injection, generation, fetching, parsing, and update database. Figure 2 shows these major components and a complete cycle of a typical web crawler. The first phase of *injection* selects the list of pre-selected seed URLs to initiate the crawling process. Next, the *generation* phase ranks the URLs according to a ranking algorithm and selects top URLs for the initial crawling. In the *fetching* stage, crawler sends

web requests to the selected URLs to fetch and store the HTML[2] of webpages. Next, the *parser* stage parses crawled webpages to extract and index the *main content* and outerlinks found on webpages. Finally, during *update database* phase new website ranking score calculated through a ranking algorithm is assigned for the selection of URLs in the next cycle.

For LRL crawling, we select open-source Apache-Nutch web crawler version 2.4 released in October, 2019 [83]. In addition, Apache Hadoop HDFS v2.7 [84] infrastructure is used with total of 10 machines, 125 GB RAM, 25 TB storage, and 45 CPU cores. Also, Hadoop database *(Hbase) v0.98* [85] is integrated with Nutch for indexing of the crawled data. As such, Nutch web crawler is not suitable for crawling of LRL web content due to three major limitations. First, the majority of the URLs on the WWW contains content of English and European languages. Therefore, the absence of any language filter will result in a low yield rate of Nutch web crawler. Yield rate is defined as the successful crawling rate per crawl cycle [86]. Second, Nutch web crawler has implemented different ranking algorithms like Online Page Importance Computation, Link Analysis, and WebGraph to calculate website ranking score of URLs in the update database phase [87]. These algorithms generally prioritize popular websites having huge number of outlinks for crawling which in many cases is not true for LRL webpages. Finally, webpages often contain HTML tags, advertisements, headers, footers etc., that need to be removed. Therefore, effective noise mitigation filters are required for Nutch web crawler to develop high-quality mono-lingual corpora.

Algorithm 1 LRL Website Scoring Algorithm

```
function Mapper(key, page)
    host ← getHost(key)
    EMIT(host, page)

function Reducer(key, PageArray)
    LangeBytes ← 0
    FetchedDocCounter ← 0
    PageCache[] ← NULL
    foreach( page in PageArray ):
    PageCache.add(page)
    if pagenotfetched then
        continue
    LangBytes += getLangBytes(page)
    FetchedDocCounter += 1
    close loop
    avg.bytes ← LangBytes/FetchedDocCounter
    foreach( page in PageCache ):
    PageScore ← avg.bytes
    EMIT(key, page)
    close loop
```

2) CONTENT PARSING

In Nutch, different plugins are available to parse various MIME[3] types like HTML, JSON,[4] JavaScript, and Zip etc. In particular, the most common MIME type of 'text/html' is parsed using HTML parser. HTML parser removes HTML tags and extracts the remaining text. However, this approach also selects the noisy text from unnecessary sections like publicity, banner, and menus etc. In this regard, first, we integrate open-source library Boilerpipe [88] in HTML parser of Nutch. Boilerpipe uses HTML tree structure of a webpage and text-based features with a binary classifier to extract the main content. In general, the Boilerpipe module of 'Article Extractor' provides significant results for the extraction of the main content. However, we observe that Boilerpipe also selects noisy text such as headers and captions along with the main content. Therefore, we enhance the Boilerpipe library by introducing a rule-based algorithm of Web-Article Miner (Web-AM) [89]. Web-AM removes the noise of Boilerpipe selected content using the observation that the main content is comprised of large length with simple formatting and noisy content contains short text with rich formatting. Web-AM scans the complete HTML tree and selects content from the only node with the maximum number of characters and its neighboring nodes at the same tree level. The noisy content on other tree levels is discarded by Web-AM. The Web-AM algorithm and its implementation details are described in our previous research [89].

3) CONTENT FILTER

Next, we find different natural languages – language distribution – present in the content extracted by Web-AM using open-source language identification library of Compact Language Detector 2 (CLD2) [42]. Our goal here is to select webpages of any target language. However, we observe that some webpages in which target language is identified contain insufficient amount of target language content. Therefore, the selection of such webpages introduces noise in the corpus. To remove such low-content noisy webpages, we design a *content-size filter* that selects the minimum number of documents contributing to the 95% of the total bytes of that target language present in the crawled data.

To design such a filter, first, we crawl webpages of pre-selected seed URLs of a target language. Next, we extract the main content of crawled webpages using Web-AM and calculate the total bytes of target language present in the crawled data. Then, we apply *minimum threshold* values of 32, 64, 128, 256, and 512 bytes of target language to remove webpages with content less than threshold value. To select appropriate threshold value for content-size filter, we compare the target language bytes present in all fetched webpages selected after each threshold value ($\text{Bytes}_{\text{threshold}}$) with total bytes ($\text{Bytes}_{\text{total}}$) present in all these webpages

[2]Hyper Text Markup Language
[3]Multipurpose Internet Mail Extensions
[4]JavaScript Object Notation

TABLE 2. Definition and description of defined terms.

SR#	Abbreviation	Referring to	Description
1	*Page*	Webpage	HTML document available at crawled URL
2	*Content*	Content of page	Main content extracted from crawled page
3	*Website*	Website	Domain of crawled URL
4	*TLG*	Target language	The low resource language considered for analysis
5	*ContentSize*	Size of content	Size of content in terms of bytes or tokens

using Equation 2

$$Percentage_{threshold} = \frac{Bytes_{\text{threshold}}}{Bytes_{\text{total}}} \times 100 \tag{2}$$

$Minimum_{threshold}$ value preserving at least 95 percent bytes of a target language is selected as threshold value of content-size filter. We integrate the Java implementation of Boilerpipe, Web-AM, and content-size filter in Nutch crawler.

4) LRL WEBSITE SCORING

As mentioned earlier, website ranking score algorithms available in Nutch are biased towards popular websites. Therefore, LRL crawling needs a language focused scoring algorithm. To achieve this goal, we define our own *LRL website scoring* algorithm to assign a ranking score to websites. The aim of our algorithm is to prioritize websites for crawling with high-quality content of the target language. To assign LRL website score to a particular website, first, we calculate the target language bytes present in each webpage of a website that is successfully crawled in previous crawling cycles. Next, the LRL score is assigned by calculating the average number of target language bytes present in all webpages of a website. Let n number of webpages from a crawled website (*website*) and each webpage contains target language bytes ($TLGBytes_{webpage}$). Then, the LRL website score is calculated by using Equation 3.

$$LRLScore_{website} = \frac{1}{n}\sum_{j=1}^{n} TLGBytes_{webpage} \tag{3}$$

The algorithm to calculate the LRL website score of websites is presented in Algorithm 1. We implement the LRL scoring algorithm in Java language and integrate it into Nutch in 'UpdateDB' class. Furthermore, we have released Nutch plugins of Web-AM, content-size filter, and LRL website scoring algorithm on SourceForge.[5] With few configurable parameters, one can crawl a high-quality language corpus of any LRL with our plugins.

B. CORPUS ANALYZER

The second major module of our Corpulyzer framework performs analysis of the developed corpus using traditional evaluation metrics used in web measurements, NLP, and IR as shown in Figure 1. The details of these evaluation metrics are provided in Section III. However, we note that these commonly used evaluation metrics do not capture the content-richness of a website with respect to the target language.

In this regard, we define new evaluation metrics of content-richness to perform in-depth analysis of the crawled corpus from different aspects.

Our content-richness measure consists of two metrics: i) webpage-language-share and ii) website-language-richness. Table 2 describes terms and abbreviations used to define new metrics.

- **Webpage-language-share** is defined as the percentage of target language content measured in bytes or tokens in a webpage. We calculate the webpage-language-share (LSh_{TLG}) score of a TLG using Equation 4.

$$LSh_{TLG} = \frac{ContentSize_{TLG}}{ContentSize_{total}} \times 100 \tag{4}$$

- **Website-language-richness** is an average value of a webpage-language-share scores of all webpages of a website. We calculate the website-language-richness (SiteR) score of a target website using Equation 5.

$$SiteR_{TLG} = \frac{1}{n}\sum_{j=1}^{n} LSh_{TLG}j \tag{5}$$

Given the traditional and newly defined metrics, corpus analyzer initiates analysis of the corpus from different angles. It is worth noting that URL depth alludes to the website structure. The proposed corpus analyzer initiates web measurements analysis of the corpus by examining the frequency of webpages from each website and their respective URL depth. Next, we focus on measuring the quality of crawled websites using the Alexa traffic rank service and Harmonic centrality ranking. For NLP/IR analysis, first, the diversity of corpus vocabulary is measured by calculating language and non-language tokens using a uni-code range of target language. Similar to vocabulary, heterogeneity of webpages is explored by detecting exact and near duplicates. For exact-duplicates detection, first, the hash value of all webpages is calculated by parsing the content to Message-Digest-5 (MD5) algorithm [90]. Then, MD5 hash value of all webpages is compared to identify *exact-duplicates* with the same hash code. Furthermore, we detect *near-duplicates* using fuzzy hash algorithm of *Textprofile-Signature* [91]. It tokenizes the textual field and selects the effective alphanumeric tokens by discarding tokens with length less than a fixed threshold. Moreover, Zipf's law that provides insight on the distribution of all tokens in the corpus is examined.

[5]https://sourceforge.net/projects/corpulyzer-urdu/

Finally, the content-richness of webpage/website is measured using our newly defined metrics.

V. URDU LANGUAGE – CASE STUDY

In this section, we present a case study of the Urdu language corpus using Corpulyzer. First, we show the efficacy of the Corpulyzer framework. Next, we describe the process of developing *UrduWeb20* corpus. Finally, we prepare other datasets from the Common Crawl Corpus for the comparison.

A. CORPULYZER – PERFORMANCE EVALUATION

The key objective of the Corpulyzer framework is to tweak the biasness of default Nutch crawler towards language-rich websites containing high-quality content of the target language. Next, we explore the performance evaluation perspective of the Corpulyzer framework. For this purpose, we run Corpulyzer with default Nutch crawler and with our modifications as described in Section IV. First, we prepare two sets of seed URLs: i) *Urdu-rich* and ii) *random*. Here, we refer webpages with at least 256 bytes of Urdu content size as Urdu-rich webpages whereas random may contain any number of Urdu bytes. The reason for using 256 bytes content size threshold is explained when we describe *UrduWeb20* dataset. In addition, for each ranking score algorithm, we crawl the WWW three times using seed URLs selected with following combinations of Urdu-rich and random sets: i) 50% URLs from both sets, ii) 25% from Urdu-rich and 75% URLs from random set, and iii) 100% URLs from the random set. The Nutch crawler is configured to crawl top 10,000 URLs for eleven cycles with total 10,000 seed URLs. After crawling, the yield rate of the Crawler for each cycle is calculated as in [41]. Let *TotalPages* be the total number of webpages successfully crawled by the crawler in a cycle and *UrduPages* are the number of crawled pages having Urdu language content. Then, the yield rate of crawler for the cycle is calculated using Equation 6

$$Yieldrate = \frac{UrduPages}{TotalPages} \times 100 \tag{6}$$

Figure 3 shows the yield rate of the crawler for default and LRL website scoring algorithms with a combination of 50% seed URLs selected from two sets. We observe that initially, for the first cycle, both crawling algorithms achieve 63% yield rate because same seed URLs are fetched in both cases. However, the yield rate in case of default crawler drops exponentially to almost 0% in the 6th cycle. Interestingly, this result shows that crawler is unable to fetch more Urdu webpages and is wasting resources on crawling non-Urdu content. On the other hand, the LRL website scoring algorithm improves the yield rate value from 63 to 73%. Moreover, average values of yield rate for 11 cycles are 13% and 70% for default and LRL website scoring algorithm, respectively. We note that after further applying content-size filter of 256 bytes after crawling, yield rate slightly dropped as expected. The crawling results using other combinations of seed URLs from the two sets show similar behavior. These results establish that Corpulyzer is an essential and effective tool for LRL crawling.

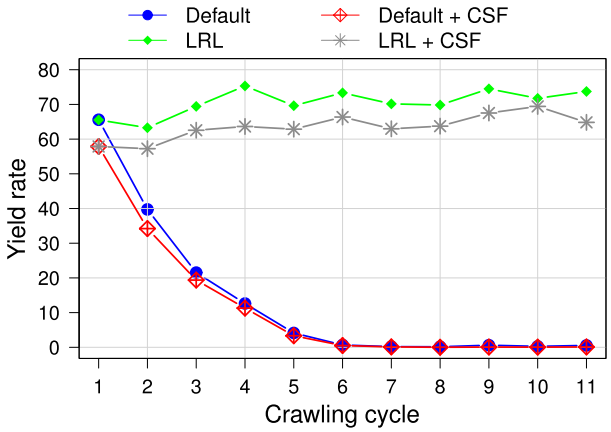


FIGURE 3. Yield rate of default and LRL website scoring with seed of 50% Urdu-rich URLs (CSF = Content-size filter).

TABLE 3. Urdu bytes vs webpages for content-size filter.

Threshold	webpages	% webpages	Bytes (MB)	% Bytes
0 Bytes	331302	100	607	100
32 Bytes	328826	99.25	604.64	99.6
64 Bytes	316646	95.57	604.04	99.5
128 Bytes	292935	88.42	601.64	99.11
256 Bytes	204813	61.82	586.41	96.48
512 Bytes	171682	51.82	572	94.23

B. UrduWeb20 CORPUS

Before availing Corpulyzer to build high-quality Urdu language corpus, we need to address one challenging question regarding the optimal threshold value for the content-size filter (Section IV) that removes noisy webpages with minimal loss and yet keeps reasonable amount of the overall crawled data. To determine the threshold value, first, we filter URLs of 1.28 million Urdu webpages from the Common Crawl release of December 2016. Our previous research [40] elaborates the implementation details and characteristics of these URLs. However, these URLs are selected after parsing the complete HTML of webpages while Corpulyzer parses only the main content of webpages extracted using Web-AM. We crawl all these URLs and filter out Urdu webpages using Web-AM and CLD2. Interestingly, only 25.7% (0.33 million) out of 1.28 million webpages are identified as containing Urdu content after parsing the main content. In order to calculate the optimal threshold value, we calculate $Bytes_{total}$ in all these crawled webpages and $Bytes_{threshold}$ after applying the threshold value of 32, 64, 128, 256, and 512 bytes. Table 3 provides the number of webpages and Urdu bytes after applying different values of thresholds. We observe that the threshold value of 256 bytes preserves 96.5% of Urdu bytes while removing 38.2% noisy webpages fulfilling the criteria of selecting the minimum number of webpages while preserving 95% Urdu content. Hence, we use the threshold

TABLE 4. Statistics of *CC-Urdu-html*, *CC-Urdu-crawl*, and *UrduWeb20* datasets.

SR#	Corpus	Duration	CC-Urdu-html			CC-Urdu-crawl		
			Size	Webapages	Websites	Size	Webapages	Websites
1	**CC16**	1-14 Dec.2016	29 GB	1.2 M	20.7K	12 GB	0.2 M	2.0 K
2	**CC18**	9-19 Dec. 2018	21 GB	7.8 M	17.7K	11 GB	0.2 M	3.4 K
3	**CC19**	16-27 Jun. 2019	27 GB	8.9 M	15.6K	13 Gb	0.3 M	4.7 K
	UrduWeb20	Jan. 2016 - Jun. 2020	-	-	-	617 GB	8.0 M	6.6 K

TABLE 5. Statistics of *CC-Urdu-meta* dataset.

			Data Size		Pages		Domains
SR#	Corpus	Duration	Total	Urdu	Total	Urdu	Urdu
1	**CC18-34**	14-22 Aug. 2018	220+ TB	32 GB	2.6 B	0.8 M	17.9 K
2	**CC18-39**	17-26 Sep. 2018	220+ TB	33 GB	2.8 B	1.4 M	19.1 K
3	**CC18-43**	15-24 Oct. 2018	240+ TB	35 GB	3.0 B	1.0 M	20.5 K
4	**CC18-47**	12-22 Nov. 2018	220+ TB	36 GB	2.6 B	1.0 M	19.0 K
5	**CC18-51**	9-19 Dec. 2018	250+ TB	23 GB	2.1 B	1.1 M	22.0 K
6	**CC19-04**	15-24 Jan. 2019	240+ TB	40 GB	2.8 B	1.0 M	21.0 K
7	**CC19-09**	15-24 Feb. 2019	225+ TB	36 GB	2.9 B	0.9 M	22.7 K
8	**CC19-13**	18-27 Mar. 2019	210+ TB	36 GB	2.6 B	0.8 M	17.7 K
9	**CC19-18**	18-26 Apr. 2019	198+ TB	33 GB	2.5 B	0.8 M	16.2 K
10	**CC19-22**	19-27 May. 2019	220+ TB	36 GB	2.6 B	0.8 M	16.9 K
11	**CC19-26**	16-27 Jun. 2019	220+ TB	29 GB	2.6 B	0.9 M	16.0 K
	Total	Aug. 2018 - Jun. 2019	2463 + TB	369 GB	29.1 B	10.9 M	209 K
	Unique	Aug. 2018 - Jun. 2019	-	158 GB	-	5.2 M	57.6 K

value of 256 bytes for the Urdu language webpages. It is worth noting that although our empirical evaluation to determine a threshold value for content-size filter is generic, however, this threshold may differ for other LRLs due to character encoding. Next, we initiate the crawling of WWW for Urdu webpages by using URLs of 1.28 million Urdu webpages mentioned above. In addition, URLs of manually selected top 1,000 Urdu websites are added in the seed. Leveraging from these seed URLs, Web-AM, content-size filter with 256 bytes threshold, and LRL website scoring algorithm, we crawl the WWW for four years from 2016 to 2020 and successfully collect 8.0 million Urdu webpages crawled from 6,590 websites. The details of UrduWeb20 corpus are given in Table 4.

C. COMMON CRAWL URDU CORPUS

Common Crawl Corpus (CCC) is commonly used to develop corpora for LRLs ([38], [40], [92], [93]) by filtering language-specific webpages instead of crawling the whole WWW indigenously. In general, CCC release meta-data as well as the crawled content where former is lightweight and easier to analyze and latter requires huge bandwidth to download and store the data. As an alternate strategy, we build three datasets using CC released data: i) CC-meta, ii) CC-Urdu-meta, and ii) CC-Urdu-crawl. First, we build *CC-meta* dataset to explore the impact of URL selection and crawling strategies of Common Crawl in general. This dataset consists of meta-information of 29.1 billion URLs in 11 common crawl releases from September2018 – June2019. This meta-information of each release is available in the form of compressed files (>200GB size) with information of webpage URL, MIME-type, and charset etc [94]. Next, we build *CC-Urdu-meta* dataset by filtering out Urdu webpages. We note that from August 2018 onward releases [95], CC also provides ISO[6] language code of top three languages present in webpages after parsing HTML of the webpage from CLD2. We select a webpage only if the Urdu language is detected irrespective of the rank provided by CLD2. Table 5 provides details of CC-Urdu-meta dataset containing URLs of Urdu webpages. For each CC release, Urdu webpages from 16K-23K websites are crawled and the cumulative size of these HTML pages remains in the range of 32-40GB. CC-Urdu-meta contains URLs of 10.9 million Urdu webpages, however, after de-duplication, it has 5.2 million unique webpages crawled from 57.6K websites. In essence, CC-Urdu-meta dataset contains only URLs of Urdu webpages. *UrduWeb20* on the other hand contains content of webpages extracted after parsing HTML of webpages from Web-AM and content-size filter. Therefore, we also crawl Urdu webpages using URLs available in CC-Urdu-meta to build *CC-Urdu-html*. Finally, we parse CC-Urdu-html from same filters to build *CC-Urdu-crawl* for fair comparison. However, due to limited resources, we crawl Urdu webpages from only three CCC releases, namely: i) December 2016 (*CC16*), ii) December 2018 (*CC18*), and iii) June 2019 (*CC19*). Table 4 provides statistics of CC-Urdu-html and CC-Urdu-crawl corpora. In general, we find that 70-76% webpages in CC-Urdu-html corpus contain only noisy content as they are removed by Web-AM and content-size filters. Additionally, upto 85% websites contain only noisy Urdu webpages. After crawling and filtration, CC16, CC18,

[6]International Organization for Standardization

TABLE 6. Names and descriptions of datasets.

Sr#	Dataset	Description
1	*UrduWeb20*	UrduWeb20 contains high-quality 8.0 million Urdu webpages crawled from WWW using Corpulyzer.
2	*CC-meta*	The dataset contains meta information of all webpages released by Common Crawl from August 2018 - June 2019
3	*CC-Urdu-meta*	The dataset contains meta information of Urdu webpage filtered from *CC-meta*.
4	*CC-Urdu-html*	The dataset contains complete html content of Urdu webpages from CC16, CC18, and CC19.
5	*CC-Urdu-crawl*	It contains Urdu webpages from *CC-Urdu-html* filtered after applying Web-AM and content-size filter.

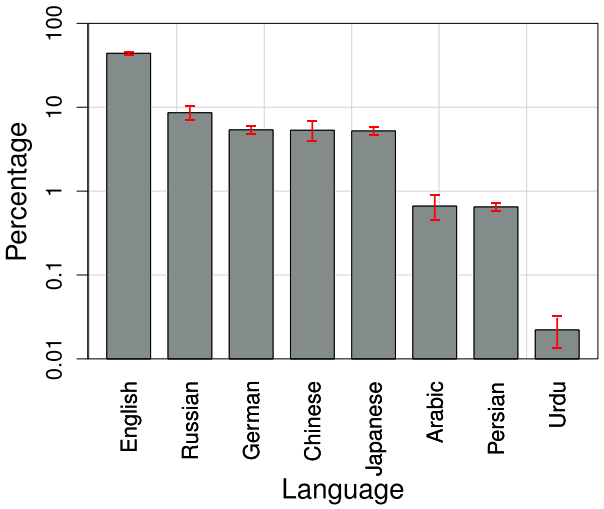


FIGURE 4. Top 5 and Perso-Arabic languages in *CC-meta*.

and CC19 have 0.2, 0.2, and 0.3 million webpages, respectively. Table 6 provides descriptions of all datasets.

VI. RESULTS

In this section, we compare different characteristics of our datasets obtained through Common Crawl and Corpulyzer. First, we present results related to web measurements. Next, we discuss content-richness of webpages and websites. Finally, we study the language diversity through the lens of NLP and IR.

A. WEB MEASUREMENTS

1) CONTENT DISTRIBUTION

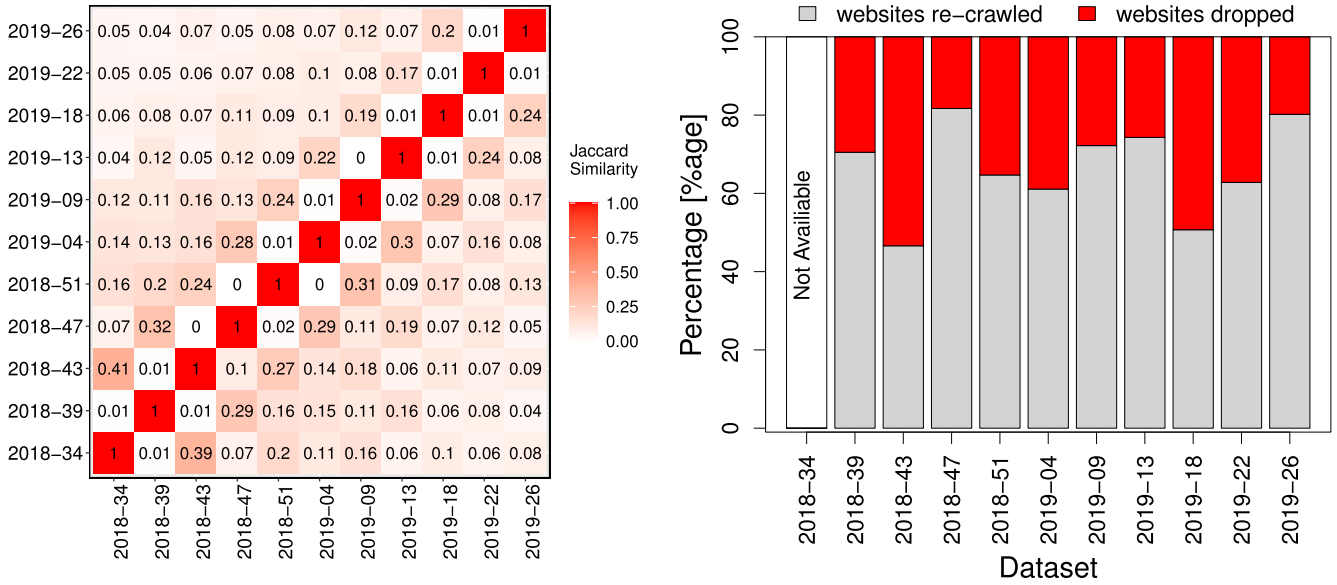
We begin our analysis by asking a question that how much of LRL content is crawled by the Common Crawl that can be effectively used to build LRL corpora. To answer this question, we examine CC-meta dataset. Figure 4 shows the mean percentage of webpages belonging to the five most-frequent languages, i.e., English, Russian, German, Chinese, and Japanese. The percentage values of Perso-Arabic languages such as Arabic, Persian, and Urdu are also provided. Unsurprisingly, English language content dominates the Common Crawl with 43.92% webpages. However, Arabic, Persian, and Urdu have a very low percentage of webpages with values of 0.66, 0.64, and 0.022%, respectively. In addition, the webpages with the MIME type of text are generally used to build textual corpora [96], [97]. Therefore, we also study the distribution of content MIME types in all the *CC-meta* dataset. We note that *text* is the most dominant MIME type with a share of 98% webpages while the percentage of image MIME type varies from 0.02% to 3%. From this result, one can conclude that crawlers of Common Crawl are designed to prioritize webpages with text content to optimize the storage and crawling bandwidth. In general, the Common Crawl contains a uniform distribution of different language webpages in different releases. We also note that *CC-meta* is dominated by the textual content of high-resource languages. These results are consistent with our previous observations reported in Shafiq *et al.* [40].

2) CRAWLING – URL SELECTION ALGORITHM

Next, we focus on exploring the URL selection algorithm of the Common Crawl by examining the diversity of webpages and websites in CC-meta dataset. First, we analyze the overlap of webpages in CC-meta by calculating the Jaccard similarity coefficient of URLs. Figure 5a illustrates the overlap of URLs in CC-meta and CC-Urdu-meta datasets where more overlap is highlighted with the dark shaded area. It is worth noting that two consecutive releases of CC have negligible overlap between URLs while alternate consecutive data points have maximum overlap. This is an interesting pattern and raises an important question regarding the selection algorithm of websites for the crawling. We investigate this question by calculating the churn-rate (Section III) of websites within two consecutive releases of CC-Urdu-meta. Figure 5b shows the percentage of websites dropped and re-crawled in subsequent releases of CC-Urdu-meta. Note that the churn rate for the first release cannot be calculated. On average, 34% websites are dropped by Common Crawl in the next release. The churn rate of websites varies from 18 to 55%. We conclude that the Common Crawl URL selection algorithm schedules URLs from different websites for re-crawling after a certain period of time to get the updated webpage.

3) CONTENT COVERAGE

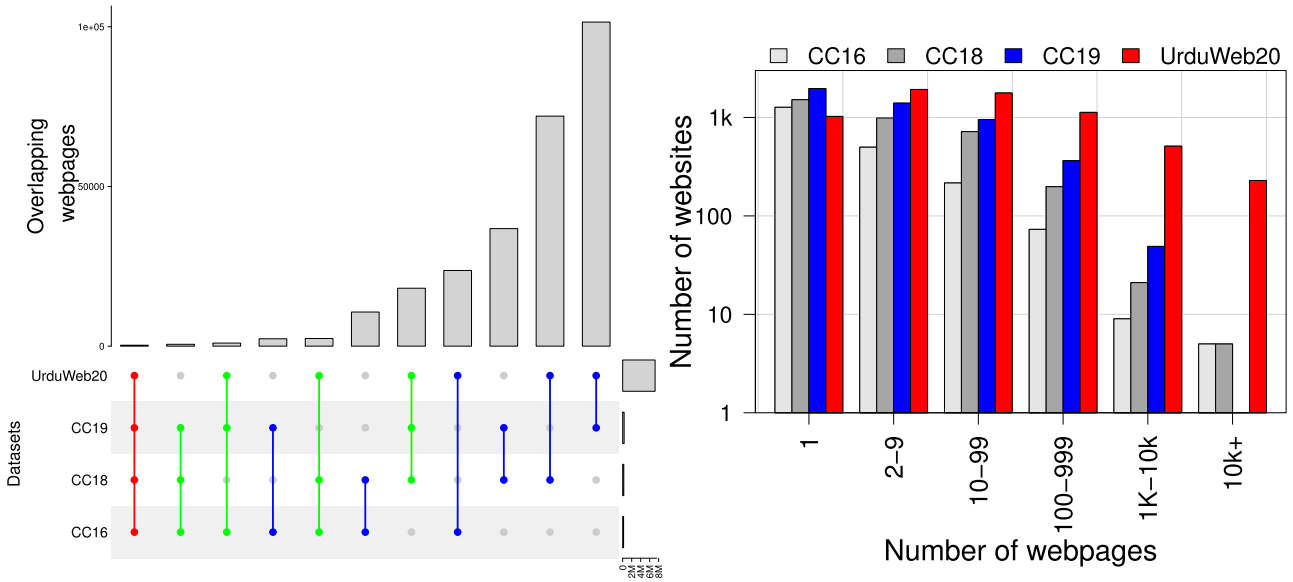
Next, we compare CC-Urdu-crawl and UrduWeb20 with the question that how many webpages in CC-Urdu-crawl are also crawled by the Corpulyzer to estimate the coverage of webpages in UrduWeb20. Figure 6a shows the number of overlapping URLs in all 11 possible combinations of CC16, CC18, CC19, and UrduWeb20 datasets. Interestingly, UrduWeb20 contains only 26K webpages from CC16. Intuitively, one may expect that UrduWeb20 should contain all



(a) Lower diagonal: *CC-meta* , upper diagonal:*CC-Urdu-meta*.

(b) Churn rate of websites in *CC-Urdu-meta*.

FIGURE 5. Common Crawl Corpus analysis for a) URLs overlap with Jaccard similarity b) churn rate of websites.

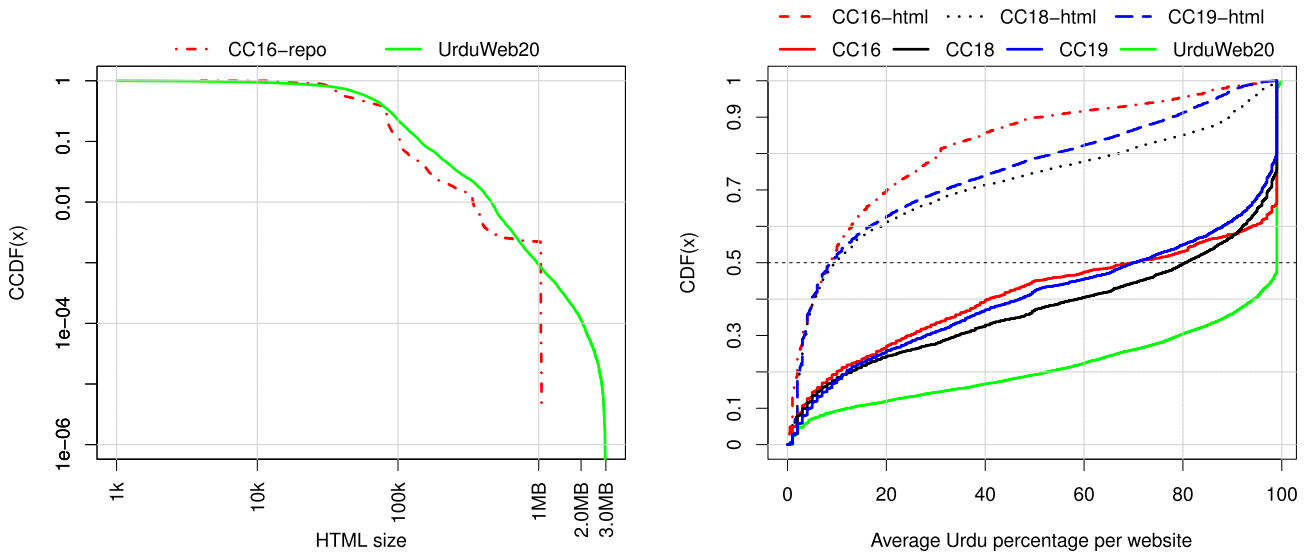


(a) Overlapping webpages.

(b) Distribution of webpages (log-scale).

FIGURE 6. Analysis of *CC-Urdu-crawl* and *UrduWeb20* for a) webpages overlap b) webpages per website.

webpages of CC16 because it was used as a seed to initiate the crawling. However, in UrduWeb20, webpages are re-crawled periodically to remove dead webpages. Hence, these dead webpages from CC16 are removed in UrduWeb20. Moreover, UrduWeb20 contains 31.2% (101K) webpages from the latest dataset of CC19. Another aspect regarding coverage of content is the extent to which the crawler crawls webpages from the websites. To inspect this aspect, Figure 6b compares the frequency of websites in UrduWeb20 and CC-Urdu-crawl with 1, 2-9, 10-99, 100-999, 1k-10k and greater than 10k webpages. We observe that 3.47% websites in UrduWeb20 and only 0.02-0.25% websites in CC-Urdu-crawl datasets contain greater than 10k webpages. Also, 40-61% websites in CC-Urdu-crawl contain only one webpage. To explore this issue further, we find the URL depth of all URLs in UrduWeb20 and CC-Urdu-crawl. We note that all URLs in CC-Urdu-crawl datasets have maximum URL depth of 12 while UrduWeb20 crawls webpages up to the URL depth of 25. We conclude that the Common Crawl crawls only limited number of webpages from a single



(a) Comparison of HTML webpage size.

(b) CDF of website-language-richness.

FIGURE 7. Comparison of Common Crawl and *UrduWeb20* for a) webpage size b) website-language-richness.

website to cover large number of websites. However, in-depth crawling of websites with high-quality LRL content is done by Corpulyzer which is highly desirable to build large scale high-quality corpus of LRL. Furthermore, we compare the size of raw HTML webpages in UrduWeb20 with Common Crawl. For this purpose, we build the *CC16-repo* dataset by downloading the HTML of Urdu webpages in CC16 from Common Crawl repository available on Amazon Simple Storage Service (Amazon S3) [98]. Figure 7a shows the CCDF plot of HTML webpage sizes of *CC16-repo* and *UrduWeb20*. We observe that both datasets have majority of webpages with size between 10 KB to 1MB. However, this plot also indicates that during crawling webpages having a size greater than 1MB are trimmed by Common Crawl to optimize storage resources [99]. We note that UrduWeb20 contains 7,354 webpages with size greater than 1MB. Our manual analysis of these webpages reveals that religious blogs and discussion forums contain large-sized HTML webpages. The performance of different NLP applications like classification, translation, and summarization will be affected adversely due to the absence of content from these domains.

B. WEB CONTENT RICHNESS

1) WEBPAGE LANGUAGE SHARE

Next, we focus our attention towards content-rich LRL webpages and websites. First, we compare webpage-language-share (LSh_{Urdu}) values of CC-Urdu-html, CC-Urdu-crawl, and UrduWeb20 to test the efficacy of different filters integrated into the Corpulyzer. Table 7 shows frequency of webpages in bins of 0-49, 50-79, 80-89, 90-94, and 95-100 LSh_{Urdu} values. For brevity, in case of CC-Urdu-html and CC-Urdu-crawl, only value of *CC18* are provided. We observe that CC-Urdu-html contains 45% webpages with less than 50% LSh_{Urdu}. In addition, only

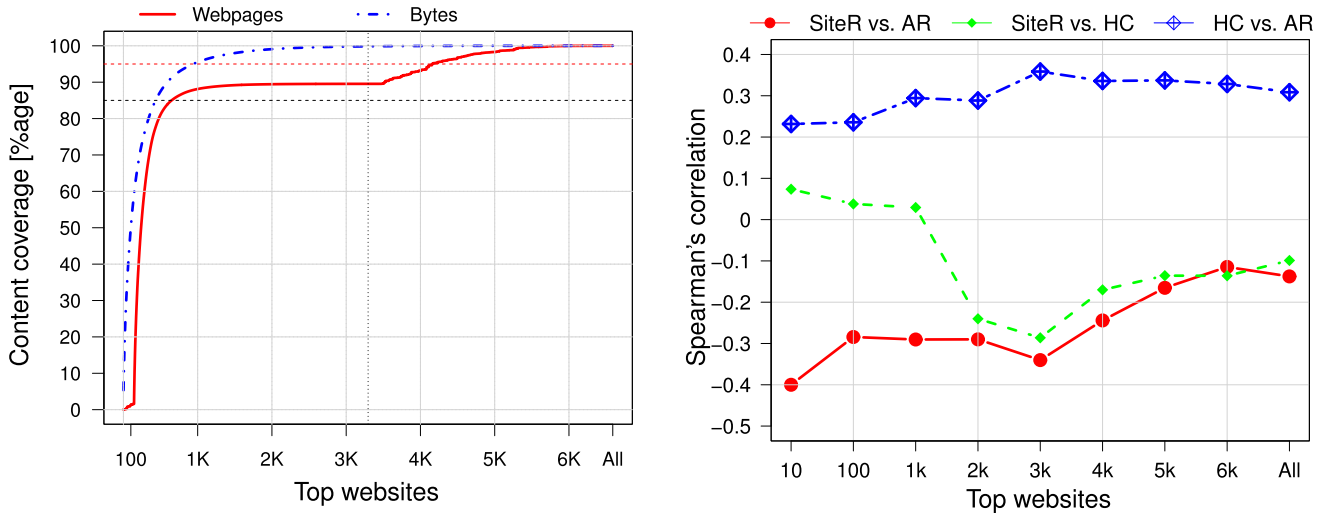
TABLE 7. Comparison of *Webpage-language-share*.

LSh_{Urdu} %	CC-Urdu-html		CC-Urdu-crawl		UrduWeb20	
	Pages	%	Pages	%	Pages	%
0-49	364539	45.66	3659	1.47	46038	0.58
50-79	179256	23.58	13157	5.29	42618	0.53
80-89	112968	14.15	2267	0.91	13522	0.17
89-94	103405	12.95	3108	1.25	8038	0.1
95-100	29062	3.64	226399	89.38	7894784	98.62
Total	789230	100	248590	100	8000506	100

16% webpages contain more than 90% Urdu content. Interestingly, UrduWeb20 and CC-Urdu-crawl contain 88-89% and 98% webpages with >95% LSh_{Urdu}. Also, CC-Urdu-crawl contains less than 9% webpages with less than 90% LSh_{Urdu}. We conclude that Web-AM and content-size filter select high-quality webpages of Urdu language. Moreover, we investigate the distribution of other languages present in different datasets. Table 8 provides the frequency of most frequent languages of Urdu, English, Arabic, and Persian present in UrduWeb20 and CC-Urdu-crawl. English is the second most common language which is present in 7-13% webpages of CC-Urdu-crawl and 21.66% webpages of UrduWeb20. We also observe that Arabic and Persian are succeeding English with the presence of up to 1.25 and 1.15% webpages in CC-Urdu-crawl and UrduWeb20, respectively. The presence of these languages is not unexpected due to the common vocabulary in Perso-Arabic languages.

2) WEBSITE CONTENT RICHNESS

At a high level, just like webpages, identification of content-rich websites is crucial for building high-quality LRL corpora. For this purpose, we plot CDF of website-language-richness ($SiteR_{Urdu}$) of websites in CC-Urdu-html, CC-Urdu-crawl, and UrduWeb20 in Figure 7b. Our analysis reveals that 50% websites have 8%, 70%, and 89% Urdu



(a) Content-coverage.

(b) Comparison of website ranking.

FIGURE 8. Comparison of top websites in *UrduWeb20* for a) *content-coverage* b) website ranking.

TABLE 8. Distribution of languages in *CC-Urdu-crawl* and *UrduWeb20*.

		CC16		CC18		CC19		UrduWeb20	
Rank	Language	Frequency	%	Frequency	%	Frequency	%	Frequency	%
1	**Urdu**	204 K	100	249 K	100	325 K	100	8005 K	100
2	**English**	15.8 K	7.75	24.9 K	10	41.6 K	12.79	1733 K	21.66
3	**Arabic**	0.9 K	0.46	2.1	0.83	4.0 K	1.25	126 K	1.58
4	**Persian**	2.1 K	1.03	1.6	0.63	2.3 K	0.71	68.7 K	0.86
	Other	4.3 K	2.13	5.6	2.26	11.1 K	3.41	388 K	4.85

content in CC-Urdu-html, CC-Urdu-crawl, and UrduWeb20, respectively. The results provide three compelling insights regarding the selection of websites for LRL content. First, CC-Urdu-html contains 69–90% of websites contain noisy Urdu content and these numbers are quite understandable because Common Crawl is generic and not language focused. Second, Web-AM and content-size filter facilitate selection of content-rich websites highlighted by the increase in median $\text{SiteR}_{\text{Urdu}}$ value by a factor of 7.7–8.4. Finally, crawling of WWW using LRL website scoring algorithm selects highly content-rich websites because average $\text{SiteR}_{\text{Urdu}}$ is further enhanced by a factor of 9–10 compared to CC-Urdu-html. The key factors of such variations in results are Corpulyzer's LRL website scoring algorithm (Section IV) and content filters that prefer websites having more Urdu content while selecting webpages for crawling.

Furthermore, in order to examine the *content-coverage* of the content-rich website in *UrduWeb20*, we explore the frequency and bytes of webpages in top-N websites according to $\text{SiteR}_{\text{Urdu}}$ score. We define *content-coverage* across top-N websites as percentage of content in terms of number of webpages and bytes crawled from these websites. Figure 8a shows the percentage of webpages and bytes contributed by top-100, 1k, 2k, 3k, 4k, 5k, 6k, and 'All' websites in UrduWeb20. We found that the top 500 (7.6%) websites account for 81% webpages and 87% bytes. Interestingly, we find that 95% of bytes in the UrduWeb20 are contributed by the top 1000 (15%) websites. This high coverage is mainly due to LRL website scoring algorithm as websites with higher number of Urdu bytes are preferred by the Corpulyzer. In addition, we analyze $\text{SiteR}_{\text{Urdu}}$ scores of 229 (3.47%) websites with 10k+ Urdu webpages (see Figure 6b) and found that these websites are ranked among top 1000 websites in the UrduWeb20. The discovery of these content-rich websites by the Corpulyzer is a valuable contribution in the perspective of LRL corpus building. Table 9 provides a list of top websites in UrduWeb20 w.r.t. number of webpages and bytes.

3) WEBSITE RANKING

Next, we compare website rankings of 6,590 websites by $\text{SiteR}_{\text{Urdu}}$ with two other well-known website ranking lists of harmonic centrality (HC) and Alexa global traffic rank (AR). For HC websites ranking, we leverage the publicly available web-graph of websites released by the Common Crawl. In particular, we use HC ranking of websites available in web-graph of May/Jun/Jul 2019 [100]. We compare rankings of $\text{SiteR}_{\text{Urdu}}$ with HC and AR using Spearman's rank correlation coefficient [101]. Figure 8b shows the correlation between $\text{SiteR}_{\text{Urdu}}$, HC, and AR rankings after selecting top 10, 100, 1k, 2k, 3k, 4k, 5k, 6k, and 'All' websites according to $\text{SiteR}_{\text{Urdu}}$ score. Surprisingly, $\text{SiteR}_{\text{Urdu}}$ ranking has a negative correlation with both Alexa and harmonic centrality

TABLE 9. Top websites in *UrduWeb20* w.r.t. webpages and Urdu bytes.

	Webpages						Urdu bytes					
SR#	Website	SiteR	Pages	%	Bytes	%	Website	SiteR	Pages	%	Bytes	%
1	islamtimes.org	99	123031	1.54	1396MB	4.5	islamtimes.org	99	123031	1.54	1396MB	4.5
2	ur.wikipedia.org	99	114260	1.43	154MB	0.5	dunyapakistan.com	99	88796	1.11	390MB	1.2
3	urdupoint.com	99	107720	1.35	207MB	0.7	bbc.com	99	86763	1.08	377MB	1.2
4	dunyapakistan.com	99	88796	1.11	390MB	1.2	uniurdu.com	92	44964	0.56	359MB	1.2
5	bbc.com	99	86763	1.08	377MB	1.2	dawnnews.tv	99	73833	0.92	342MB	1.1
6	javedch.com	99	86033	1.08	195MB	0.6	daleel.pk	99	38632	0.48	308MB	0.9
7	jasarat.com	99	82584	1.03	224MB	0.7	dailyazadiquetta.com	99	75281	0.94	275MB	0.8
8	dailyazadiquetta.com	99	75281	0.94	275MB	0.8	mukaalma.com	99	37149	0.46	270MB	0.8
9	dawnnews.tv	99	73833	0.92	342MB	1.1	mazameen.com	99	23818	0.3	264MB	0.8
10	urdu.news18.com	99	69986	0.87	155MB	0.5	dailypakistan.pk	99	56470	0.71	264MB	0.8
	Total		908287	11.35	3719MB	11.9			648737	8.1	4250 MB	13.6

TABLE 10. Statistics of vocabulary in *CC-Urdu-crawl* and *UrduWeb20*.

Dataset	Unique tokens			Total tokens		
	Total	Urdu	Non-Urdu	Total	Urdu	Non-Urdu
CC16	1,567,443	1,135,445 (72.44%)	431,998 (27.56%)	83,041,289	739,866,25 (89.10%)	9,054,664 (10.90%)
CC18	1,657,531	1,170,334 (70.61%)	487,197 (29.39%)	110,478,335	99,414,292 (89.99%)	11,064,043 (10.01)
CC19	2,291,104	1,559,268 (68.06%)	731,836 (31.94%)	180,230,789	156,170,722 (86.65%)	24,060,067 (13.35%)
UrduWeb20	13,099,438	8,453,207 (64.53%)	4,664,231 (35.47%)	4,117,611,602	3,695,875,215 (89.75%)	421,736,387 (10.25)

ranking ranging from -0.1 to -0.4. $\text{SiteR}_{\text{Urdu}}$ has a high negative correlation with AR and HC when top 3K websites are selected. These top 3k websites cover 97% Urdu bytes and 89% Urdu webpages in UrduWeb20, highlighted in Figure 8a. From these result, we conclude that generic website ranking is not effective for LRL corpus building.

C. NLP/IR

1) VOCABULARY DISTRIBUTION

Vocabulary distribution is an eminent feature to measure the diversity of the corpus. As mentioned in Section III, we calculate *unique*, *total*, *language*, and *non-language* tokens using unicode range of Urdu '*U+0600 to U+06FF, U+0750 to U+077F, U+FB50 to U+FDFF, and U+FE70 to U+FEFF*' [102]. Table 10 shows the distribution of language and non-language tokens for UrduWeb20 and CC-Urdu-crawl. The results indicate the similar distribution of language and non-language tokens in CC-Urdu-crawl and UrduWeb20 with 27-35% unique and 10-13% total non-Urdu tokens. We further analyze the vocabulary of dataset using Zipf's law. Figure 9 shows the Zipf's law distribution of UrduWeb20 for 13.1, 110.3 and 360.6 million unique unigram, bi-gram, and tri-gram tokens, respectively. Overall, the vocabulary distributions in UrduWeb20 conforms to Zipf's law.

2) DUPLICATE WEBPAGES

Similar to NLP, the performance of IR systems heavily rely on the diversity of webpages in the corpus. The presence of duplicate content impacts the quality of search results. In general, CC-Urdu-crawl dataset contains 8-16% exact and 13-23% near duplicates. Similarly, UrduWeb20 contains 12% exact and 8% near duplicates. We manually examined these

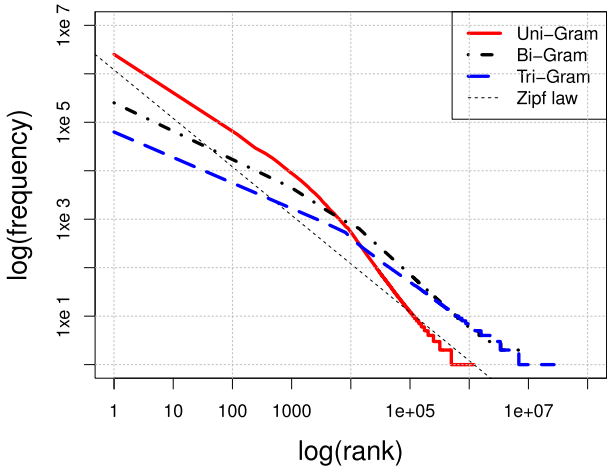


FIGURE 9. Zipf's law distribution of *UrduWeb20*.

duplicate webpages in detail and found that in some case webpages are labelled as duplicates due to website archival practice. Webmasters archived their webpages on different URLs and crawlers fetched both webpages resulting in duplicate webpages. Our results highlight the importance of using de-duplication algorithms to build the LRL corpora.

VII. UrduWeb20 – COMPARISON AND APPLICATIONS

In this section, first we compare different characteristics of *UrduWeb20* with other Urdu language corpora. Next, we present NLP and IR applications built by using *UrduWeb20*.

A. COMPARISON WITH OTHER DATASETS

Recently, the research community has focused on the development of Urdu language corpus to build various applications such as topic classification, sentiment analysis, fake news

TABLE 11. Comparison of *UrduWeb20* with Urdu language corpora (Web = Web measurements, CR = Content-richness).

SR#	Name	Purpose	Pages	Tokens	Availability		Corpus Analysis		
					Public	URL	Web	NLP/IR	CR
1	Northwestern Polytechnical University Urdu (NPUU) [103]	Topic classification	10,819	3,611,756	Yes	No	No	Yes	No
2	COrpus of Urdu News TExt Reuse (COUNTER) [104]	Topic classification	1,200	288,835	Yes	No	Yes	Yes	No
3	Naive collection [105]	Topic classification	5,003	2,216,845	No	-	No	Yes	No
4	Collection of Urdu News Text (COUNT19) [62]	Topic classification	10,451	91,840	No	-	No	Yes	No
5	Urdu Corpus [106]	Topic classification	26,067	19,296,846	No	-	No	Yes	No
6	DSL Urdu News [107]	Topic classification	662	130,000*	No	-	No	Yes	No
7	Bend the truth corpus [108]	Fake news identification	9,00	20,572	Yes	No	No	Yes	No
8	Urdu Paraphrase Plagiarism Corpus (UPPC) [109]	Plagiarism detection	160	6,201	Yes	No	No	Yes	No
9	Common Crawl Urdu [40]	Large scale repository	1,277,591	1,250,000,000*	No	-	Yes	Yes	Yes
	Urdu Web 20 (UrduWeb20)	Large scale repository	8,000,506	4,117,611,602	Yes	Yes	Yes	Yes	Yes

* Not the exact number of tokens. Reported in paper

detection, plagiarism detection, and different IR systems. Table 11 summarizes different characteristics of other available corpora and UrduWeb20. For instance, Northwestern Polytechnical University Urdu (NPUU) and COrpus of Urdu News TExtReuse (COUNTER) are publicly available Urdu language datasets build to train classifiers for the news text classification. However, only NLP/IR based characteristics of vocabulary distribution of news for NPUU (10,819) and COUNTER (1,200) are available. Similarly, Naive collection, Collection of Urdu NewsText (COUNT19), Urdu Corpus, and DSL Urdu News datasets have been developed to train classifiers for Urdu text classification. These datasets contain a limited number of webpages (662–26,067) due to limited scope of developed applications. For fake news detection, a dataset of 'bend the truth corpus' containing only 900 Urdu news webpages is released publicly. Similarly, for plagiarism detection, Urdu Paraphrase Plagiarism Corpus (UPPC) dataset of 160 articles is developed. Additionally, large scale Urdu dataset of 'Common Crawl Urdu' was developed. The dataset was built by filtering 1.28 million Urdu webpages from the Common Crawl. Our survey highlights two distinct characteristics of *UrduWeb20*. First, the majority of Urdu datasets are built for specific tasks like text classification and they contain a limited number of webpages to reduce the human effort. On the other hand, *UrduWeb20* provides a large amount of Urdu content which can be easily used to develop corpora for various applications. Second, the in-depth characteristics of web measurements, NLP/IR, and content-richness of UrduWeb20 are examined to make it a valuable resource for the research communities in the NLP/IR fields.

B. NLP/IR APPLICATIONS

UrduWeb20 is effectively used to develop and test NLP and IR applications for the Urdu language. For instance, UrduWeb20 is employed by Kausar *et al.* [61] for the propaganda detection from the Urdu content. Authors train machine learning models on the gold standard dataset of Urdu content. For validation of models, 6.4 million Urdu webpages from our previous release of UrduWeb20 are classified. Authors pre-process the text by removing special characters and URLs from the text. Next, the Urdu stop words are removed using a manually defined dictionary of Urdu stopwords. Finally, state-of-the-art classification models with multiple text features are evaluated to find the best performing model and features. In addition, the propaganda score of each website is calculated to identify websites spreading malicious content in the Urdu language. Similarly, an IR application '*Parakh*'[7] is built to detect the plagiarized content in the Urdu language. This system uses UrduWeb20 along with Urdu language theses and research papers to detect and generate the plagiarism report. Furthermore, selected webpages of UrduWeb20 are also used to develop Urdu domain classification system [62]. We believe UrduWeb20 is one step forward in removing obstacles such as scarcity of high-quality data faced by the NLP/IR research community.

VIII. CONCLUSION

A high-quality textual corpus is a pre-requisite to build state-of-the-art AI-based services. However, for Low-Resource Languages (LRL), the development of these services face the major challenge of corpora scarcity. In this article, we purpose ''Corpulyzer'' – a novel framework for building low resource language corpora using our proposed LRL website scoring and content-size filter. In addition, our framework analyzes crawled corpus using metrics from web measurements, NLP/IR, and content-richness. In particular, we introduce two metrics of webpage-language-share and website-content-richness to measure the content richness of LRL corpora. Using Corpulyzer, we prepare a Urdu language web corpus (UrduWeb20) containing 8 million webpages crawled from 6,590 websites. Another contribution of this article is in-depth comparison of *UrduWeb20* with three Urdu language corpora filtered from Common Crawl Corpus (CCC). Our

[7] https://parakh.cle.org.pk/

results show that Corpulyzer improves the average yield rate of the crawler from 13% to 70%. Moreover, the selection of websites with content-size filter enhances the value of website-content-richness by a factor of 7.7-8.4. Also, top 3,000 websites selected according to website-content-richness cover 97% Urdu content of UrduWeb20. Our framework is generic and can be used for the development of large scale corpora for different natural languages.

In future, we plan to enhance Corpulyzer framework further by integrating the URL scheduling algorithm along with LRL websites scoring for extensive coverage of LRL content on WWW. In addition, we will use different optimization strategies for the selection of the threshold value of the content-size filter to offset different character encoding formats for different LRLs. With regards to content-richness metrics, other complex content quality metrics such as readability, cohesion, and coherence can be included for the corpus evaluation.

REFERENCES

[1] N. Zengeler, T. Kopinski, and U. Handmann, "Hand gesture recognition in automotive human-machine interaction using depth cameras," *Sensors*, vol. 19, no. 1, p. 59, Dec. 2018.

[2] V. Suma, "Computer vision for human-machine interaction-review," *J. Trends Comput. Sci. Smart Technol.*, vol. 1, no. 2, pp. 131–139, 2019.

[3] M. P. Aylett, B. R. Cowan, and L. Clark, "Siri, echo and performance: You have to suffer darling," in *Proc. Extended Abstr. CHI Conf. Hum. Factors Comput. Syst.*, May 2019, pp. 1–10.

[4] M. Soleymani, D. Garcia, B. Jou, B. Schuller, S.-F. Chang, and M. Pantic, "A survey of multimodal sentiment analysis," *Image Vis. Comput.*, vol. 65, pp. 3–14, Sep. 2017.

[5] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 8, no. 4, p. e1253, 2018.

[6] L. Ran, L. Zheng, L. Hailun, W. Weiping, and M. Dan, "Text emotion analysis: A survey," *J. Comput. Res. Develop.*, vol. 55, no. 1, p. 30, 2018.

[7] D. Gupta, "Study on extrinsic text plagiarism detection techniques and tools," *J. Eng. Sci. Technol. Rev.*, vol. 9, no. 5, pp. 1–15, 2016.

[8] M. H. Al-Bayed and S. S. Abu-Naser, "Intelligent multi-language plagiarism detection system," *Int. J. Acad. Inf. Syst. Res.*, vol. 2, no. 3, pp. 19–34, 2018.

[9] R. M. Orr, G. R. Nell, and B. L. Brumbaugh, "Intelligent assistant for home automation," U.S. Patent 10 170 123, Jan. 1, 2019.

[10] F.-L. Li, M. Qiu, H. Chen, X. Wang, X. Gao, J. Huang, J. Ren, Z. Zhao, W. Zhao, L. Wang, G. Jin, W. Chu, "Alime assist: An intelligent assistant for creating an innovative e-commerce experience," in *Proc. CIKM*, 2017, pp. 2495–2498.

[11] S. Reshmi and K. Balakrishnan, "Empowering chatbots with business intelligence by big data integration," *Int. J. Adv. Res. Comput. Sci.*, vol. 9, no. 1, pp. 1–5, 2018.

[12] R. Singh, M. Paste, N. Shinde, H. Patel, and N. Mishra, "Chatbot using TensorFlow for small businesses," in *Proc. 2nd Int. Conf. Inventive Commun. Comput. Technol. (ICICCT)*, Apr. 2018, pp. 1614–1619.

[13] D. Diefenbach, A. Both, K. Singh, and P. Maret, "Towards a question answering system over the semantic Web," *Semantic Web*, vol. 11, no. 3, pp. 421–439, Apr. 2020.

[14] A. Abdi, N. Idris, and Z. Ahmad, "QAPD: An ontology-based question answering system in the physics domain," *Soft Comput.*, vol. 22, no. 1, pp. 213–230, Jan. 2018.

[15] W. Cui, Y. Xiao, and W. Wang, "KBQA: An online template based question answering system over freebase," in *Proc. IJCAI*, vol. 16, 2016, pp. 9–15.

[16] L. Wu, D. Hu, L. Hong, and H. Liu, "Turning clicks into purchases: Revenue optimization for product search in e-commerce," in *Proc. SIGIR Conf. Res. Develop. Inf. Retr.*, 2018, pp. 365–374.

[17] M. Zia and R. C. Rao, "Search advertising: Budget allocation across search engines," *Marketing Sci.*, vol. 38, no. 6, pp. 1023–1037, 2019.

[18] M. Zhou, Z. Ding, J. Tang, and D. Yin, "Micro behaviors: A new perspective in e-commerce recommender systems," in *Proc. WSDM*, 2018, pp. 727–735.

[19] A. Greenstein-Messica and L. Rokach, "Personal price aware multi-seller recommender system: Evidence from eBay," *Knowl.-Based Syst.*, vol. 150, pp. 14–26, Jun. 2018.

[20] M. H. Nguyen. (2020). *Chatbot Market 2020: Stats, Trends, Size and Ecosystem Research*. Accessed: Oct. 8, 2020. [Online]. Available: https://www.businessinsider.com/chatbot-market-stats-trends

[21] Market and Markets. (May 2018). *Natural Language Generation (NLG) Market*. Accessed: Sep. 8, 2020. [Online]. Available: https://www.marketsandmarkets.com/Market-Reports/natural-language-generation-market-14328817.html

[22] P. Cadwell, S. O'Brien, and E. DeLuca, "More than tweets: A critical reflection on developing and testing crisis machine translation technology," *Transl. Spaces*, vol. 8, no. 2, pp. 300–333, Nov. 2019.

[23] Y.-A. Chung, W.-H. Weng, S. Tong, and J. Glass, "Towards unsupervised speech-to-text translation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 7170–7174.

[24] Ontotext. (Nov. 2019). *Text Analytics for Enterprise Use*. Accessed: Oct. 10, 2020. [Online]. Available: https://www.ontotext.com/knowledgehub/white-paper/ta-for-enterprise-use/

[25] K. Kuligowska, "Commercial chatbot: Performance evaluation, usability metrics and quality standards of embodied conversational agents," *Professionals Center Bus. Res.*, vol. 2, no. 2, pp. 1–16, 2015.

[26] F. Guzmán, P.-J. Chen, M. Ott, J. Pino, G. Lample, P. Koehn, V. Chaudhary, and M. Ranzato, "The flores evaluation datasets for low-resource machine translation: Nepali–English and Sinhala–English," in *Proc. EMNLP-IJCNLP*, 2019, pp. 6100–6113.

[27] Ethnologue. (2020). *Ethnologue*. Accessed: Oct. 10, 2020. [Online]. Available: https://www.ethnologue.com/guides/ethnologue200

[28] D. Lyons. (Jun. 2017). *How Many People Speak English, and Where is it Spoken?* Accessed: Oct. 10, 2020. [Online]. Available: https://www.babbel.com/en/magazine/how-many-people-speak-english-and-where-is-it-spoken

[29] M. Ahmed, C. Dixit, R. E. Mercer, A. Khan, M. R. Samee, and F. Urra, "Multilingual corpus creation for multilingual semantic similarity task," in *Proc. LREC*, 2020, pp. 4190–4196.

[30] M. Jakubíček, V. Kovář, P. Rychlỳ, and V. Suchomel, "Current challenges in Web corpus building," in *Proc. WAC*, 2020, pp. 1–4.

[31] P. Giampieri, "The Web as corpus and online corpora for legal translations," *Comparative Legilinguistics*, vol. 33, pp. 35–56, Feb. 2019.

[32] D. Hawking, E. Voorhees, N. Craswell, and P. Bailey, "Overview of the trec-8 Web track," in *Proc. TREC*, 1999, pp. 1–18.

[33] P. Bailey, N. Craswell, and D. Hawking, "Engineering a multi-purpose test collection for Web retrieval experiments," *Inf. Process. Manage.*, vol. 39, no. 6, pp. 853–871, Nov. 2003.

[34] M. Karimzadeh and A. MacEachren, "GeoAnnotator: A collaborative semi-automatic platform for constructing geo-annotated text corpora," *ISPRS Int. J. Geo-Inf.*, vol. 8, no. 4, p. 161, Mar. 2019.

[35] C. Luo, Y. Zheng, Y. Liu, X. Wang, J. Xu, M. Zhang, and S. Ma, "SogouT-16: A new Web corpus to embrace ir research," in *Proc. SIGIR Conf. Res. Develop. Inf. Retr.*, 2017, pp. 1233–1236.

[36] R. Schäfer, "Processing and querying large Web corpora with the COW14 architecture," in *Proc. Challenges Manage. Large Corpora (CMLC)*, 2015, p. 28.

[37] R. Suwaileh, M. Kutlu, N. Fathima, T. Elsayed, and M. Lease, "ArabicWeb16: A new crawl for today's arabic Web," in *Proc. SIGIR Conf. Res. Develop. Inf. Retr.*, 2016, pp. 673–676.

[38] I. Habernal, O. Zayed, and I. Gurevych, "C4Corpus: Multilingual Web-size corpus with free license," in *Proc. LREC*, 2016, pp. 914–922.

[39] Common Crawl Foundation. (2020). *Common Crawl Corpus*. Accessed: Oct. 10, 2020. [Online]. Available: https://commoncrawl.org/

[40] H. M. Shafiq, B. Tahir, and M. A. Mehmood, "Towards building a urdu language corpus using common crawl," *J. Intell. Fuzzy Syst.*, vol. 39, no. 2, pp. 2445–2455, Jun. 2020.

[41] H. Shafiq and M. A. Mehmood, "NCL-Crawl: A large scale language-specific Web crawling system," in *Proc. 7th Conf. Lang. Technol. (CLT)*, 2020, pp. 1–7.

[42] Jariesa and I. I. Giuliani. (Jun. 2019). *Compact Language Detector 2 (CLD2)*. Accessed: Oct. 10, 2020. [Online]. Available: https://github.com/CLD2Owners/cld2

[43] X. Zheng, T. Zhou, Z. Yu, and D. Chen, "URL rule based focused crawler," in *Proc. IEEE Int. Conf. e-Bus. Eng.*, Oct. 2008, pp. 147–154.

[44] Q. Rajput, "Ontology based semantic annotation of urdu language Web documents," *Procedia Comput. Sci.*, vol. 35, pp. 662–670, Jan. 2014.
[45] A. Vaswani, S. Bengio, and E. Brevdo, "Tensor2Tensor for neural machine translation," in *Proc. Conf. AMTA*, 2018, pp. 193–199.
[46] Y. Wang, L. Wang, M. Rastegar-Mojarad, S. Moon, F. Shen, N. Afzal, S. Liu, Y. Zeng, S. Mehrabi, S. Sohn, and H. Liu, "Clinical information extraction applications: A literature review," *J. Biomed. Informat.*, vol. 77, pp. 34–49, Jan. 2018.
[47] Y. Liu and M. Lapata, "Text summarization with pretrained encoders," in *Proc. EMNLP-IJCNLP*, 2019, pp. 3721–3731.
[48] G. Grefenstette and J. Nioche, "Estimation of English and non-English language use on the WWW," in *Proc. Recherchee Inf. Assistee par Ordinateur (RIAO)*, 2000, pp. 237–246.
[49] H. Veisi, M. MohammadAmini, and H. Hosseini, "Toward kurdish language processing: Experiments in collecting and processing the AsoSoft text corpus," *Digit. Scholarship Humanities*, vol. 35, pp. 176–193, Feb. 2019.
[50] M. A. Mehmood, H. M. Shafiq, and A. Waheed, "Understanding regional context of World Wide Web using common crawl corpus," in *Proc. IEEE 13th Malaysia Int. Conf. Commun. (MICC)*, Nov. 2017, pp. 164–169.
[51] J. Dunn, "Mapping languages: The corpus of global language use," *Lang. Resour. Eval.*, vol. 54, pp. 1–20, Apr. 2020.
[52] G. Wenzek, M.-A. Lachaux, A. Conneau, V. Chaudhary, F. Guzmán, A. Joulin, E. Grave, "CCNet: Extracting high quality monolingual datasets from Web crawl data," in *Proc. LREC*, 2020, pp. 4003–4012.
[53] A. Infrabot. (2019). *TextProfileSignature*. Accessed: Oct. 10, 2020. [Online]. Available: https://wiki.apache.org/solr/TextProfileSignature
[54] R. Hassanian-Esfahani and M.-J. Kargar, "Sectional MinHash for near-duplicate detection," *Expert Syst. Appl.*, vol. 99, pp. 203–212, Jun. 2018.
[55] E. T. Loiacono, R. T. Watson, and D. L. Goodhue, "WebQual: A measure of website quality," *Marketing Theory Appl.*, vol. 13, no. 3, pp. 432–438, 2002.
[56] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the Web," Stanford InfoLab, Stanford, CA, USA, Tech. Rep. 66, 1999. [Online]. Available: http://ilpubs.stanford.edu:8090/cgi/export/422/BibTeX/ilprints-eprint-422.bib
[57] R. Sharapov and E. Sharapova, "The problem of fuzzy duplicate detection of large texts," in *Proc. CEUR Workshop*, vol. 2212, 2018, pp. 270–277.
[58] M. A. Mehmood, A. Feldmann, S. Uhlig, and W. Willinger, "We are all treated equal, aren't we?—Flow-level performance as a function of flow size," in *Proc. IFIP Netw. Conf.*, 2014, pp. 1–9.
[59] M. A. Mehmood, N. Sarrar, S. Uhlig, and A. Feldmann, "Understanding flow performance in the wild," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2013, pp. 1410–1415.
[60] J. Wang, J. Peng, and O. Liu, "A classification approach for less popular webpages based on latent semantic analysis and rough set model," *Expert Syst. Appl.*, vol. 42, no. 1, pp. 642–648, Jan. 2015.
[61] S. Kausar, B. Tahir, and M. A. Mehmood, "ProSOUL: A framework to identify propaganda from online urdu content," *IEEE Access*, vol. 8, pp. 186039–186054, 2020.
[62] S. A. Hamza, B. Tahir, and M. A. Mehmood, "Domain identification of urdu news text," in *Proc. 22nd Int. Multitopic Conf. (INMIC)*, Nov. 2019, pp. 1–7.
[63] R. Alshammari, "Arabic text categorization using machine learning approaches," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 3, pp. 30–226, 2018.
[64] A. Panchenko, E. Ruppert, S. Faralli, S. P. Ponzetto, and C. Biemann, "Building a Web-scale dependency-parsed corpus from CommonCrawl," in *Proc. LREC*, 2018, pp. 1–8.
[65] L. Martin, B. Muller, P. J. O. Suárez, Y. Dupont, L. Romary, É. V. de la Clergerie, D. Seddah, and B. Sagot, "CamemBERT: A tasty French language model," 2019, *arXiv:1911.03894*. [Online]. Available: http://arxiv.org/abs/1911.03894
[66] H. Le, L. Vial, J. Frej, V. Segonne, M. Coavoux, B. Lecouteux, A. Allauzen, B. Crabbé, L. Besacier, and D. Schwab, "FlauBERT: Unsupervised language model pre-training for French," in *Proc. LREC*, 2020, pp. 2479–2490.
[67] S. Parida and P. Motlicek, "Abstract text summarization: A low resource challenge," in *Proc. EMNLP-IJCNLP*, 2019, pp. 5996–6000.
[68] H. Veisi, M. MohammadAmini, and H. Hosseini, "Toward kurdish language processing: Experiments in collecting and processing the asosoft text corpus," *Digit. Scholarship Humanities*, vol. 35, no. 1, pp. 176–193, 2020.
[69] S. Roziewski and W. Stokowiec, "Languagecrawl: A generic tool for building language models upon common-crawl," in *Proc. LREC*, 2016, pp. 2789–2793.
[70] J. Krasselt, P. Dressen, M. Fluor, C. Mahlow, K. Rothenhäusler, and M. Runte, "Swiss-AL: A multilingual Swiss Web corpus for applied linguistics," in *Proc. LREC*, 2020, pp. 4145–4151.
[71] J. K. Kummerfeld, S. R. Gouravajhala, J. J. Peper, V. Athreya, C. Gunasekara, J. Ganhotra, S. S. Patel, L. C. Polymenakos, and W. Lasecki, "A large-scale corpus for conversation disentanglement," in *Proc. Meeting ACL*, 2019, pp. 3846–3856.
[72] R. Abbott, B. Ecker, P. Anand, and M. Walker, "Internet argument corpus 2.0: An SQL schema for dialogic social media and the corpora to go with it," in *Proc. LREC*, 2016, pp. 4445–4452.
[73] S. Khalifa, N. Habash, D. Abdulrahim, and S. Hassan, "A large scale corpus of Gulf Arabic," in *Proc. LREC*, 2016, pp. 4282–4289.
[74] P. Fafalios, V. Iosifidis, E. Ntoutsi, and S. Dietze, "TweetsKB: A public and large-scale RDF corpus of annotated tweets," in *Proc. Eur. Semantic Web Conf.* Cham, Switzerland: Springer, 2018, pp. 177–190.
[75] M. Potthast, T. Gollub, K. Komlossy, S. Schuster, M. Wiegmann, E. Patricia, G. Fernandez, M. Hagen, and B. Stein, "Crowdsourcing a large corpus of clickbait on Twitter," in *Proc. 27th Int. Conf. Comput. Linguistics*, 2018, pp. 1498–1507.
[76] M. Diab, N. Habash, O. Rambow, M. Altantawy, and Y. Benajiba, "COLABA: Arabic dialect annotation and processing," in *Proc. LREC Workshop Semitic Lang. Process.*, 2010, pp. 66–74.
[77] A. Pasha, M. Al-Badrashiny, M. Altantawy, N. Habash, M. Pooleery, O. Rambow, R. Roth, and M. Diab, "Dira: Dialectal arabic information retrieval assistant," in *Proc. IJCNLP, Syst. Demonstrations*, 2013, pp. 13–16.
[78] N. Ljubešic, D. Fišer, and T. Erjavec, "TweetCaT: A tool for building Twitter corpora of smaller languages," in *Proc. LREC*, 2014, pp. 1–5.
[79] M. Cieliebak, J. M. Deriu, D. Egger, and F. Uzdilli, "A Twitter corpus and benchmark resources for German sentiment analysis," in *Proc. SocialNLP*, 2017, pp. 45–51.
[80] Shuyo. (Dec. 2015). *Language Detection*. Accessed: Oct. 10, 2020. [Online]. Available: https://github.com/shuyo/language-detection/blob/wiki/ProjectHome.md
[81] L. Tan, H. Zhang, C. Clarke, and M. Smucker, "Lexical comparison between wikipedia and Twitter corpora by using word embeddings," in *Proc. Int. Joint Conf. Natural Lang. Process.*, 2015, pp. 657–661.
[82] M. G. Noll and C. Meinel, "Exploring social annotations for Web document classification," in *Proc. Symp. Appl. Comput.*, 2008, pp. 2315–2320.
[83] M. Cafarella and D. Cutting, "Building Nutch: Open source search," *Queue*, vol. 2, no. 2, pp. 54–61, Apr. 2004.
[84] T. White, *Hadoop: The Definitive Guide*. Sebastopol, CA, USA: O'Reilly Media, 2012.
[85] M. N. Vora, "Hadoop-HBase for large-scale data," in *Proc. ICCSNT*, vol. 1, 2011, pp. 601–605.
[86] V. Suchomel and J. Pomikálek, "Efficient Web crawling for large text corpora," in *Proc. WAC*, 2012, pp. 39–43.
[87] A. Infrabot. (May 2019). *Nutch Scoring*. Accessed: Oct. 10, 2020. [Online]. Available: https://cwiki.apache.org/confluence/display/NUTCH/NutchScoring
[88] C. Kohlschütter, P. Fankhauser, and W. Nejdl, "Boilerplate detection using shallow text features," in *Proc. 3rd ACM Int. Conf. Web Search Data Mining (WSDM)*, 2010, pp. 441–450.
[89] N. Aslam, B. Tahir, H. M. Shafiq, and M. A. Mehmood, "Web-AM: An efficient boilerplate removal algorithm for Web articles," in *Proc. Int. Conf. Frontiers Inf. Technol. (FIT)*, Dec. 2019, pp. 287–2875.
[90] R. Rivest and S. Dusse, *The MD5 Message-Digest Algorithm*, document RFC 1321, MIT Laboratory for Computer Science, 1992.
[91] D. Shahi, *Apache Solr: A Practical Approach to Enterprise Search*. New York, NY, USA: Apress, 2015.
[92] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning word vectors for 157 languages," in *Proc. LREC*, 2018, pp. 1–5.
[93] J. R. Smith, H. Saint-Amand, M. Plamada, P. Koehn, C. Callison-Burch, and A. Lopez, "Dirt cheap Web-scale parallel text from the common crawl," in *Proc. ACL*, 2013, pp. 1374–1383.
[94] E. Kilfeather. (Jun. 2019). *CDX Server API*. Accessed: Oct. 10, 2020. [Online]. Available: https://github.com/webrecorder/pywb/wiki/CDX-Server-API#api-reference

[95] S. Nagel. (Aug. 2018). *August Crawl Archive Introduces Language Annotations*. Accessed: Oct. 10, 2019. [Online]. Available: https://commoncrawl.org/2018/08/august-2018-crawl-archive-now-available/
[96] M. Baroni and M. Ueyama, "Building general- and special-purpose corpora by Web crawling," in *Proc. 13th NIJL Int. Symp., Lang. Corpora, Their Compilation Appl.*, 2006, pp. 31–40.
[97] A. Kilgarriff, S. Reddy, J. Pomikálek, and P. Avinesh, "A corpus factory for many languages," in *Proc. LREC*, 2010, pp. 1–7.
[98] M. R. Palankar, A. Iamnitchi, M. Ripeanu, and S. Garfinkel, "Amazon S3 for science grids: A viable solution?" in *Proc. Int. Workshop Data-Aware Distrib. Comput.*, 2008, pp. 55–64.
[99] S. Merity. (Jul. 2015). *Common Crawl Apache Nutch Group*. Accessed: Oct. 10, 2020. [Online]. Available: https://groups.google.com/d/msg/common-crawl/6b7g1gf912Y/ ZUrqgbIsmy4J
[100] R. Meusel, S. Vigna, O. Lehmberg, and C. Bizer, "Graph structure in the Web—Revisited: A trick of the heavy tail," in *Proc. Conf. World Wide Web*, 2014, pp. 427–432.
[101] J. H. Zar, "Significance testing of the spearman rank correlation coefficient," *J. Amer. Stat. Assoc.*, vol. 67, no. 339, pp. 578–580, Sep. 1972.
[102] M. Ijaz and S. Hussain, "Corpus based Urdu lexicon development," in *Proc. Conf. Lang. Technol. (CLT)*, vol. 73, 2007, pp. 1–12.
[103] M. P. Akhter, Z. Jiangbin, I. R. Naqvi, M. Abdelmajeed, A. Mehmood, and M. T. Sadiq, "Document-level text classification using single-layer multisize filters convolutional neural network," *IEEE Access*, vol. 8, pp. 42689–42707, 2020.
[104] M. Sharjeel, R. M. A. Nawab, and P. Rayson, "COUNTER: Corpus of urdu news text reuse," *Lang. Resour. Eval.*, vol. 51, no. 3, pp. 777–803, Sep. 2017.
[105] T. Zia, M. P. Akhter, and Q. Abbas, "Comparative study of feature selection approaches for urdu text categorization," *Malaysian J. Comput. Sci.*, vol. 28, no. 2, pp. 93–109, 2015.
[106] A. R. Ali and M. Ijaz, "Urdu text classification," in *Proc. 6th Int. Conf. Frontiers Inf. Technol. (FIT)*, 2009, pp. 1–7.
[107] M. N. Asim, M. U. Ghani, M. A. I. S. Ahmad, W. Mahmood, and A. Dengel, "Benchmark performance of machine and deep learning based methodologies for urdu text document classification," 2020, *arXiv:2003.01345*. [Online]. Available: http://arxiv.org/abs/2003.01345
[108] M. Amjad, G. Sidorov, A. Zhila, H. Gómez-Adorno, I. Voronkov, and A. Gelbukh, "Bend the truth: Benchmark dataset for fake news detection in urdu language and its evaluation," *J. Intell. Fuzzy Syst.*, pp. 1–13, Jun. 2020.
[109] M. Sharjeel, P. Rayson, and R. M. A. Nawab, "UPPC-urdu paraphrase plagiarism corpus," in *Proc. LREC*, 2016, pp. 1832–1836.

BILAL TAHIR received the B.Sc. degree in electrical engineering from the National University of Computer and Emerging Sciences (FAST-NU), Lahore, Pakistan, in 2014, and the M.S. degree in computer engineering from the University of Engineering and Technology, Lahore, in 2018. Since 2017, he has been working as a Research Officer with the Al-Khawarizmi Institute of Computer Science (KICS), UET, Lahore. His research interests include machine learning for images and text, natural language processing, deep learning, and information retrieval.

MUHAMMAD AMIR MEHMOOD received the Ph.D. degree in engineering from the Department of Electrical Engineering and Computer Science, Technische Universität Berlin/Deutsche Telekom Innovation Laboratories, Berlin, Germany, in 2012. He is currently working as an Associate Professor with the Al-Khawarizmi Institute of Computer Science, University of Engineering and Technology, Lahore, Pakistan. He has been the Director of the High Performance Computing and Networking Laboratory (HPCNL), since 2013. His research interests include Internet measurements, big data, cloud computing, information retrieval, and deep learning.