Progress Report

1. Hiring

Qualified personnel have been hired/assignedto the following positions as per the requirements detailed in the project proposal:

1. **PI**
   Dr. GhalibA. Shah. Associate Professor, KICS, UET Lahore
2. **Co-PI**
   Dr. SarmadHussain. Professor and Head, Center for Language Engineering, KICS, UET Lahore
3. **Consultant**
   Dr. Kashif Javed. Associate Professor, Department of Electrical Engineering, UET, Lahore.
4. **Team Lead (IM)**
   PhD student with experience of Natural Language Processing for Urdu.
5. **NLP**
   PhD student with a focus on Natural Language Processing and Machine Learning.
6. **SE (2)**
   Fresh graduates with past projects involving use of Machine Learning tools and techniques.
7. **NLP Linguistic Researcher**
   Linguistwith experience of creating linguistic resources for Urdu.
8. **Team Lead (CI)**
   Cloud infrastructure expert for design and operation of Urdu Search Engine infrastructure.
9. **AWS Support Engineer and Crawler Development Engineer**.
   Fresh graduates with good knowledge of search engine crawling software.
10. **Team Lead (SM)**
    Information retrieval expert for managing search engine query management.
11. **Information Retrieval**
    Two experts with in depth knowledge of information retrieval
12. **Software Engineers (SE)**
    Two fresh graduates with Java knowledge

2. Challenges

Urdu Search engine project is an inter-disciplinary effort where expertise of different research areas is pooled to build a large system. There are mainly three teams of the project, i) Cloud Infrastructure (CI), ii) Information Management (IM), and Search Management (SM). At the early stage of the project, all teams were required to select appropriate hardware and software tools and devise an integration plan of these tools. There were number of lengthy team meetings in order to finalize different hardware and software components. Next, we briefly describe the main challenges of different teams.

1. **Cloud Infrastructure (CI)**
   The first major task of this team is to select different software versions of Apache Nuch, Hadoop, and Solr that are compatible with each other. For this purpose, after extensive experimentation, the team has selected the following software versions of the following open source softwares.

a. Apache hadoop 1.2.1
b. Apache hbase 0.94.14
c. Apache Nutch 2.3.1
d. Apache Hive 1.0.0
e. Apache Solr 4.10.3

In addition, CI team has finalized the infrastructure design of Urdu Search Engine prototype. There are three main components of this design, mainly, front-end, back-end and storage as shown in Figure 1. The design is further consists of different layers of softwares across three different layers of infrastructure, platform, and storage as shown in Figure2 and 3. The front-end web server is protected with Suricata Intrusion Protection System (IPS) and in order to take care of the high volume traffic a load balancer (HAproxy) is planned. In Figure 3, compute nodes (C5 to C15) have been used for search management and map-reduce jobs. Furthermore, different software components as mentioned above have been assigned to particular hardware compute nodes. Finally, we have also assigned either local or HDFS storage to these software components.

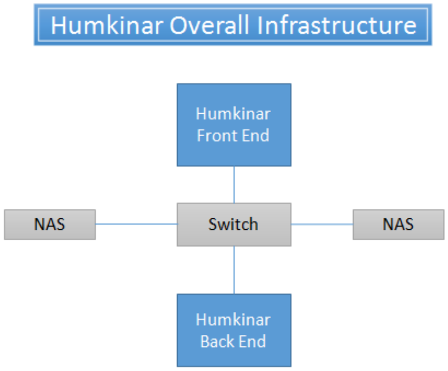


Figure 1: Urdu Search Engine infrastructure design.

The main challenge for the CI team is to make a reliable and scalable infrastructure for Urdu Search Engine. For this purpose, we selected Urdu websites by manually inspecting the content and use them as seed. Our goal is to select web sites having more than 80% Urdu content from various different categories such as web, news, poetry, blogs, and web sites of national importance.

Once the seed is decided it is challenging to run Apache Nutch crawler on multiple machine to speed up crawling process. However, the basic Nutch software can only be run as a single instance which is a big hurdle to gather large data from the web. CI team is now developing a mechanism for distributed crawling in the Nutch crawler. In addition, Apache Solr was required to index crawled documents by Nutch in respective fields, therefore the CI team designed Solr schema as shown in the Figure 6. For example, for the crawled documents in Nutch, fields such as "title", "content", "URL", and "timestamp" should also be present in the Solr. Web crawling of documents can easily eat up all the storage present in the infrastructure, therefore, CI team needs to index and store only those fields in Solr where search query is going to be executed, e.g., web content field.

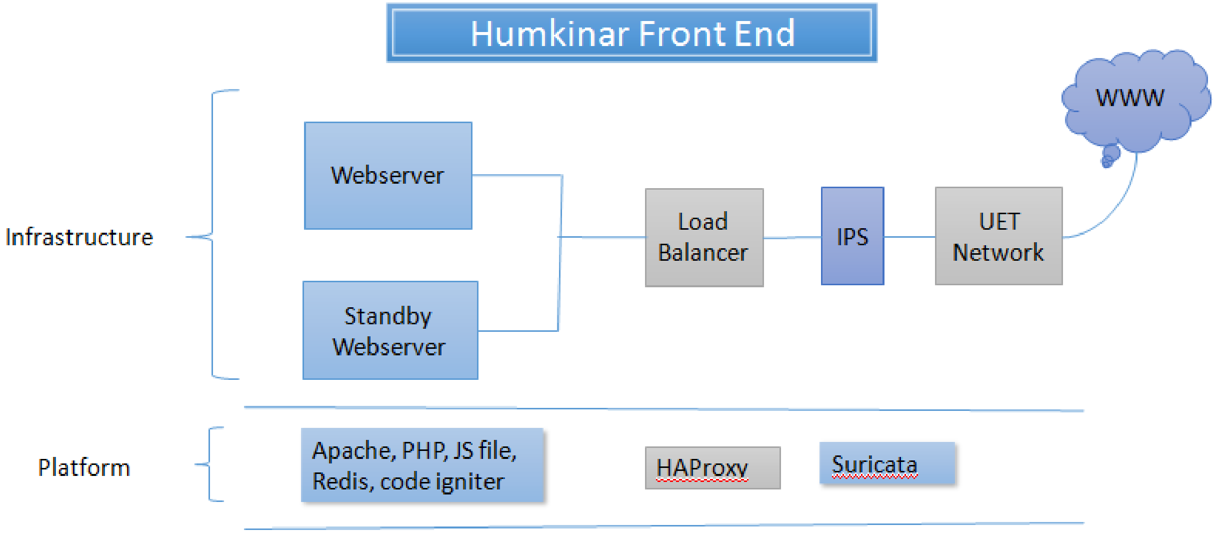


Figure 2: Urdu Search Engine Front End

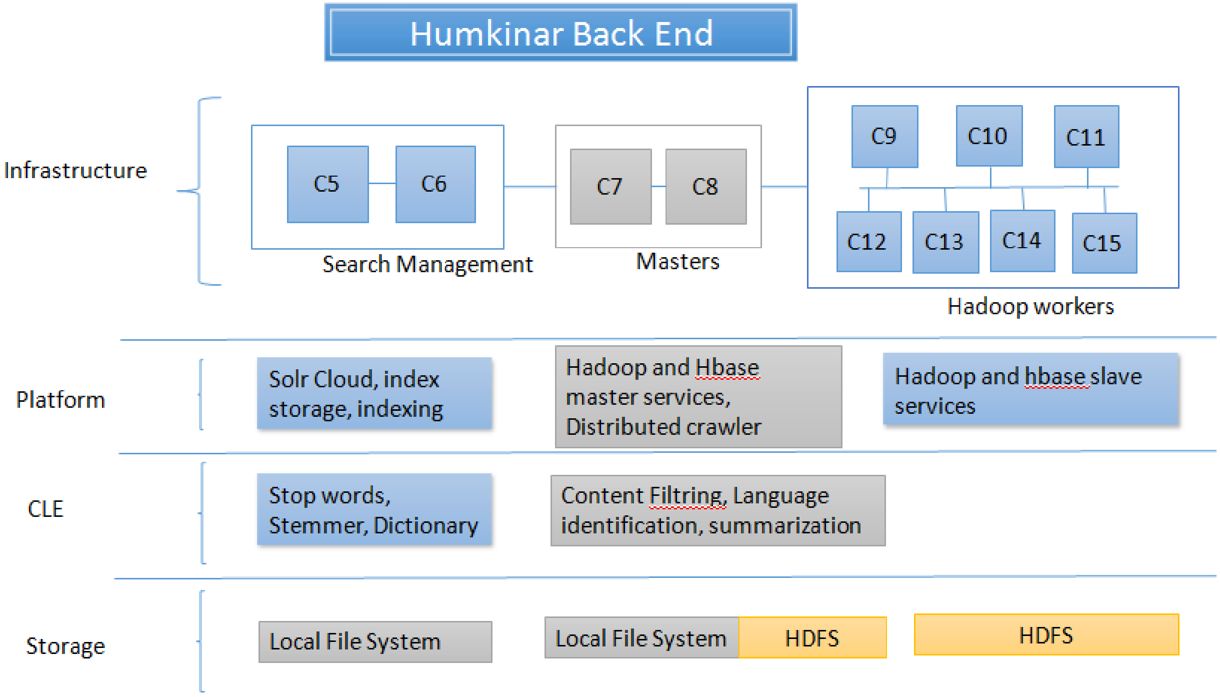

Humkinar Back End
Infrastructure
C5
C6
Search Management
C7
C8
Masters
C9
C10
C11
C12
C13
C14
C15
Hadoop workers
Platform
Solr Cloud, index storage, indexing
Hadoop and Hbase master services, Distributed crawler
Hadoop and hbase slave services
CLE
Stop words, Stemmer, Dictionary
Content Filtring, Language identification, summarization
Storage
Local File System
Local File System
HDFS
HDFS

Figure 3: Urdu Search Engine Back End.

In addition, another major challenge with respect to the storage is to tackle the growth of index over time. For this purpose, CI teams figured out how to partition index data by using different "Shards" in the Apache Solr. It was found that single core in Solr can store about 2 billion maximum documents. Although, CI team has tested Solr with multiple Shards, however, currently Solr is running with single Shard.

2. Information Management (IM)

The primary challenge faced during team setup was the availability of people with either past experience of or a history of work related to Distributed Computing and Natural Language Processing (NLP). This is largely due to the small number of domain experts available in the country and the subsequent lack of courses offered on the subject. The major focus then during the hiring phase was to find people with the right background and skills required to be an effective part of the team for this project. The Co-PI and Consultant have extensive experience of Natural Language Processing and Machine Learning, respectively. The Team Lead (IM) and NLP researcher are both PhD students who have previously worked on different projects at CLE and whose own theses center around Natural Language Processing for Urdu. The two SE hires have also done undergraduate projects using Machine Learning.

The IM team is primarily responsible for three major modules: language identification, content filtering, and text summarization. The challenges faced in the development of each of the three are in these early stages are follows:

- Language Identification:
  There is a lack of corpora of multilingual webpages that contain content in languages such as English or Arabic paired with Urdu for the validation and testing of language identification techniques. Therefore, it falls upon us to not only develop efficient and accurate methodologies for the estimation of the proportion of Urdu content in webpage, but also collect and annotate a corpus containing examples of webpages with bilingual content which we can use to evaluate the performance of our system upon. Additionally, accurate estimation of the proportion of a particular language's content remains a tough problem, especially so for Urdu due to its rich morphology which means that a particular root word may take many forms depending upon its tense, which makes classification unseen examples difficult.

- Content Filtering:
  The process of content filtering is an especially tough one due to the highly subjective nature of the task. It is quite difficult to define exactly what constitutes objectionable content, let alone devise ways of detecting it in real-world webpages. The primary job in the development of this filter was then to chalk out its scope, decide what kinds of content is to be flagged, and then devising ways of spotting such content. There is also the issue that language is very complex, and it can be quite difficult to determine whether some given content is offensive purely on the basis of a few features, since phrases can be structured in a way so as to disguise their negativity.

- Text Summarization:
  The purpose of this module is to generate summaries of webpages for viewing on mobile devices with a shortage of screen real-estate as well as to allow the user to quickly learn the gist of websites without having to go through their them in their entirety. For this too, there exists no vetted text summarization corpus of appropriate size that can be used for the development of summarization methodologies. The team is currently working on having summaries generated at different levels of a large number of articles from different domains for this very purpose. For the development of the summarization module itself, modern systems for popular languages use a number of advanced NLP

utilities in order to extract meaningful words and phrases from a body of text. Since this is one of the first efforts to do the same for the Urdu language, we must take the responsibility of developing these as well.

An overview of the three modules is given in Figure 4, followed by Figure 5 depicting the flow of information through each.

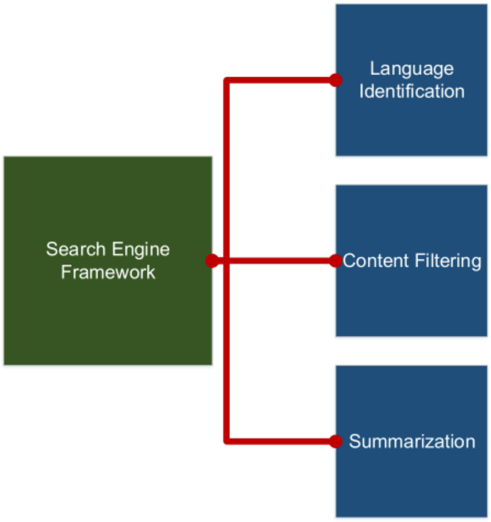


Figure 4: NLP module overview.

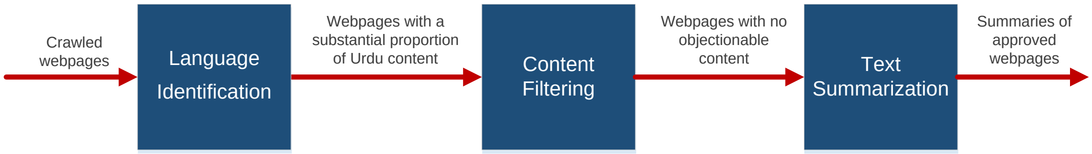

Crawled webpages
Language Identification
Webpages with a substantial proportion of Urdu content
Content Filtering
Webpages with no objectionable content
Text Summarization
Summaries of approved webpages

Figure 5: Data flow between NLP modules.

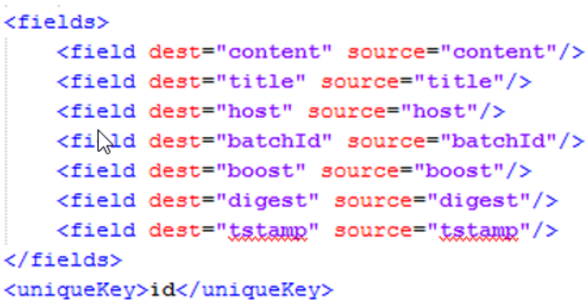
3. Search Management (SM)

This team is looking in to different open source tools for building a user friendly front-end of Urdu search engine and its related ranking algorithms for ranking the data.Initially, a web front-end was designed in core PHP along with Javascript. However, we faced cross site scripting (XSS) security problems and our site got hacked by the hackers. Therefore, in order to secure our web front-end from malicious users we selected "CodeIgniter" PHP framework to protect it from XSS and other attacks. Our new framework is more efficient, fast, and flexible. It follows Model, View, and Controller (MVC) architecture where "Model" provides data access, "View" provides rendering capabilities and "Controller"is responsible for controlling the application logic and acts as the coordinator between the View and the Model. . Moreover, due to CodeIgniter's "Model" module we have provided a mechanism of secure communication between front-end web server and back-end Apache Solr.

Another major challenge faced by the SM team is to render the content across various different web browsers, e.g., Chrome, FireFox, Safari, etc., for different screen sizes on different platforms such as Desktops, Laptops, Mobiles. To provide cross browser responsiveness, we used Bootstrap3 framework for all screen sizes and different browsers. Due to this framework, the content is automatically rendered according to the screen resolution.

With respect to Urdu language SM team faced multiple problems such as proper Urdu font selection and string manipulation. Although English language characters are easily available as ASCII characters, however, Urdu characters are available as Unicode Transformation Format (UTF-8). For our purpose, initially we selected "NafeesNastaleeq" Urdu font however by using this font some normal Urdu characters were not rendered properly. We fixed this issue by using "JameelNooriNastaleeq" font. For string manipulation, we used multi-byte string (mb_string) in PHP.

Query input in Urdu language requires Urdu language key-board. For this purpose, we have selected Open source javascript Urdu virtual keyboard for direct typing through keyboard and indirect typing through mouse. In addition, we used Persian javascript API for Urdu pagination. Figure 7 shows the front-end web page of Humkinar Pakistan.



```
<fields>
    <field dest="content" source="content"/>
    <field dest="title" source="title"/>
    <field dest="host" source="host"/>
    <field dest="batchId" source="batchId"/>
    <field dest="boost" source="boost"/>
    <field dest="digest" source="digest"/>
    <field dest="tstamp" source="tstamp"/>
</fields>
<uniqueKey>id</uniqueKey>
```

Figure:6 Code snippet of Apache Solr Schema

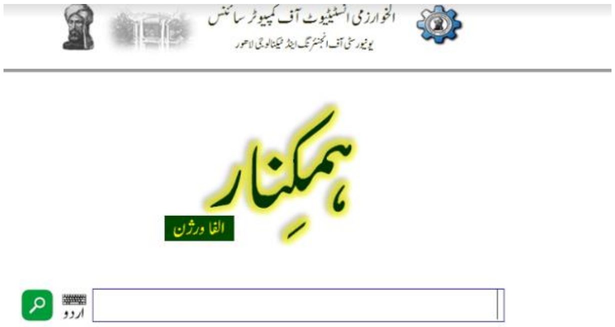

الخوارزمی انسٹیٹیوٹ آف کمپیوٹر سائنس
یونیورسٹی آف انجنیئرنگ اینڈ ٹیکنالوجی لاہور
ہمکنار
الفا ورژن
اردو

Figure: 7: Urdu Search Engine front-end

3. Trainings

All team members were put through a rigorous four-day crash course in Natural Language Processing as part of the CLE Postgraduate Certificate in Computational Linguistics, held on 26, 28 May and 3, 4 June, 2016. The course, taught by Prof. Dr. SarmadHussain (also the Co-PI of this project), served as a broad introduction to the field and covered the following topics:

1. **Character Level**
    a. Writing Systems
    b. Unicode Encoding
    c. Word Segmentation
2. **Word Level**
    a. Lexicons
    b. Spell Checking
    c. Morphological Processing
3. **Phrase Level**
    a. Parts of Speech and its tagging
    b. Grammar and its parsing

This was important since there is little material available online on Natural Language Processing of low-resource South Asian languages. All participants were also provided with handouts to practice the taught techniques on in order to gain an in-depth understanding of their workings. The entire exercise was intended to bring the team up to speed onNatural Language Processing techniques for Urdu so they may be able to hit the ground running upon commencement of the project. Topics like Unicode encoding and language modelling, at both the word- and character-level, and lexicons became of use right at the onset of preliminary work for this project. More advanced topics covered later on in the course are sure to come handy later on during work on this project.

Similarly, CI and SM team were provided the following trainings.

1. Apache Hadoop in localmode
2. Apache Hadoop in pseudo distributed mode
3. Apache Hbase integration with Hadoop and understanding Hbase query language
4. Apache Hive integration with Hadoop and understanding Hive query language
5. Exporting Hbase table to hive as an external table
6. Apache Solr configuration and learning its query language
7. Apache Nutch integration with Hadoop, crawling few websites and indexing crawled docs to solr.