IIT ROPAR

KNOWLEDGE MARKUP LANGUAGE

PROJECT REPORT

SOCIAL NETWORKS

Mentor:
Amit Kumar Verma

Authors:
Nitin Gandhi
Paras Kumar

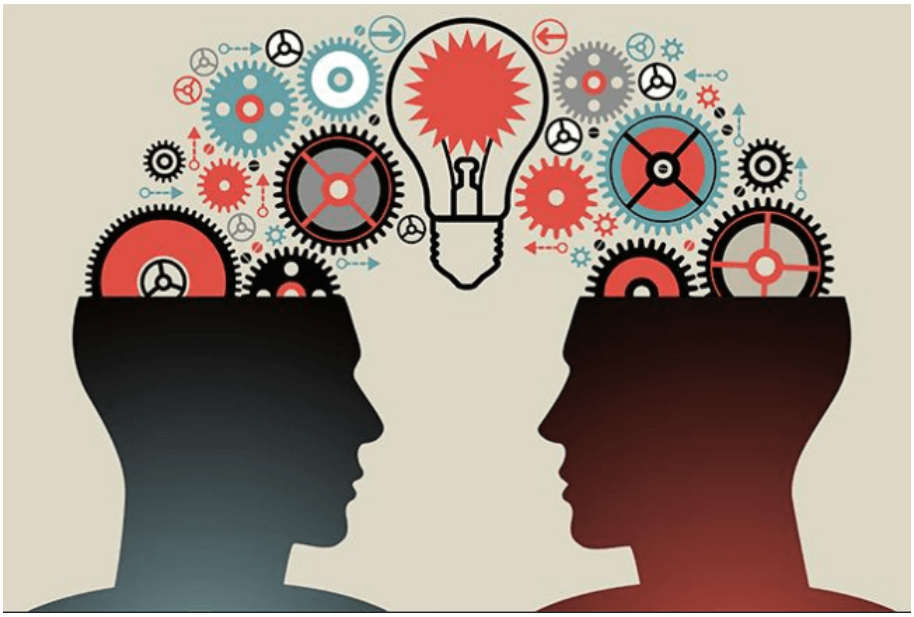
November 30, 2018

Abstract

KML stands for Knowledge Markup Language, a standard format for storing the data of all the Knowledge Building Portals. Knowledge Building portals like Wikipedia, Stack Exchange, GitHub, e.t.c provides their data dump in their own formats. We are trying to propose a new standard format for all these types of portals such that the analysis is easy.

1 Need for KML

All these Knowledge Building portals provide their data dumps in their own format. For example, Wikipedia provides its data in an XML format with their own schema definition. Similarly, Stack Exchange provides its data dump in an XML format with different schema definition. The KML will be a new standard format for these kinds of Knowledge Building portals with a standard schema definition.



2 KML Schema

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/X
  <xs:element name="KML">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="KnowledgeData">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Instance">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:int" name="RevisionId"/>
                    <xs:element type="xs:int" name="ParentId"/>
                    <xs:element name="TimeStamp">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element type="xs:dateTime" name="CreationDate"/>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="Contributors">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element type="xs:string" name="LastEditorUserName"/>
                          <xs:element type="xs:int" name="LastEditorUserId"/>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="Body">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="Text">
                            <xs:complexType>
                              <xs:simpleContent>
                                <xs:extension base="xs:string">
                                  <xs:attribute type="xs:string" name="Type"/>
                                </xs:extension>
                              </xs:simpleContent>
                            </xs:complexType>
                          </xs:element>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                  <xs:attribute type="xs:byte" name="Id"/>
                  <xs:attribute type="xs:string" name="InstanceType"/>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
            <xs:attribute type="xs:string" name="Type"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

3 Final Components of this project

1. KML-Compressor-Decompressor
2. KML-for-Github
3. Github-Spider
4. Github-Spider-Proxy-Rotated
5. User-Agent-Spider
6. Wiki-Satck-KML-Downloader

.....

Analysis	Update README.md	4 hours ago
Github-Spider-Proxy-Rotated	Create design_document.doc	5 days ago
Github-Spider	Create design_document.doc	5 days ago
KML-Compressor-Decompressor	Added code for analysis of KML	a day ago
KML-for-Github	Update README.md	5 days ago
Report	Adding updated report in pdf format	5 days ago
User-Agent-Spider	Create design_document.doc	5 days ago
compressed-KML	Create README.md	a day ago
resources	Added code for analysis of KML	a day ago
rotation_user_addresses	Rename rotation_user.txt to rotation_useragents.txt	5 days ago
schema	Create readme.md	5 days ago
updated_github_scraped_result	output of user_agent spider	5 days ago
wiki-stack-KML-Downloader	Update README.md	an hour ago
README.md	Update README.md	an hour ago

3.1 KML-Compressor-Decompressor

These programs compresses and decompresses a KML file using diff algorithm inspired from git. It can compress the KML file by 30% or more depending upon the edits made on an article. The compressed KML is also very easy to read, since the number of lines is reduced from 1 lakh to 9k and it retains the original structure of the KML so that it remains redable even when compressed.

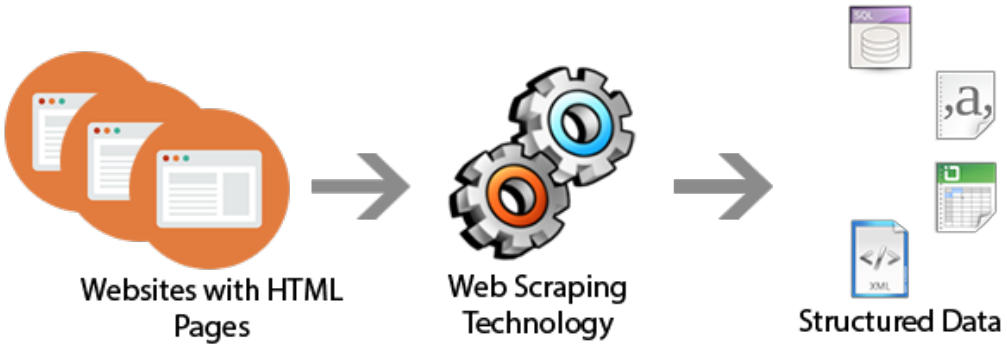
```
diff --git a/app.js b/app.js
new file mode 100644
index 0000000..144ec7f
--- /dev/null
+++ b/app.js
@@ -0,0 +1 @@
+module.exports = require('./lib/index');
diff –git a/index.js b/index.js
deleted file mode 100644
index 144ec7f..0000000
--- a/index.js
+++ /dev/null
@@ -1 +0,0 @@
-module.exports = require('./lib/index');
```

3.2 KML-for-Github

We had to make KML for github and we had KML for wikipedia earlier. The attempt of this program is to retain the structure of KML for wkipedia while representing all the data in a git workflow in this newly generated KML file. We achived this by viewing each commit as a revision of an article on wikipedia. The revisions are not compressed unlike KML for wikipedia because the commits can be highly unrelated.

3.3 Github-Spider

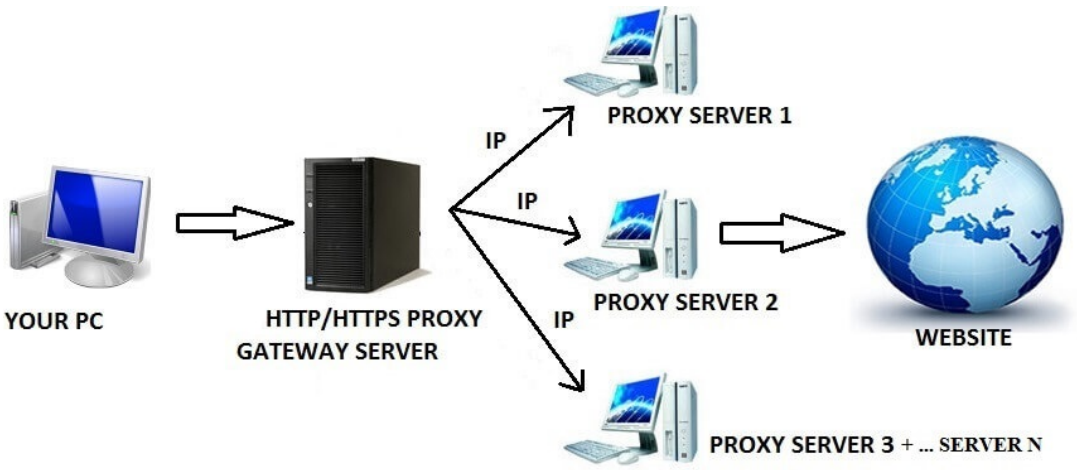
This multithreaded web-spider follows the rules of /robots.txt file and is useful for fetching small sized GitHub repositories. It outputs the results in JSON format.



This spider searches the webpage using a simple Breadth-First Search Algorithm for extracting the information from each link which is obtained using the corresponding XPath addressing path-forms.

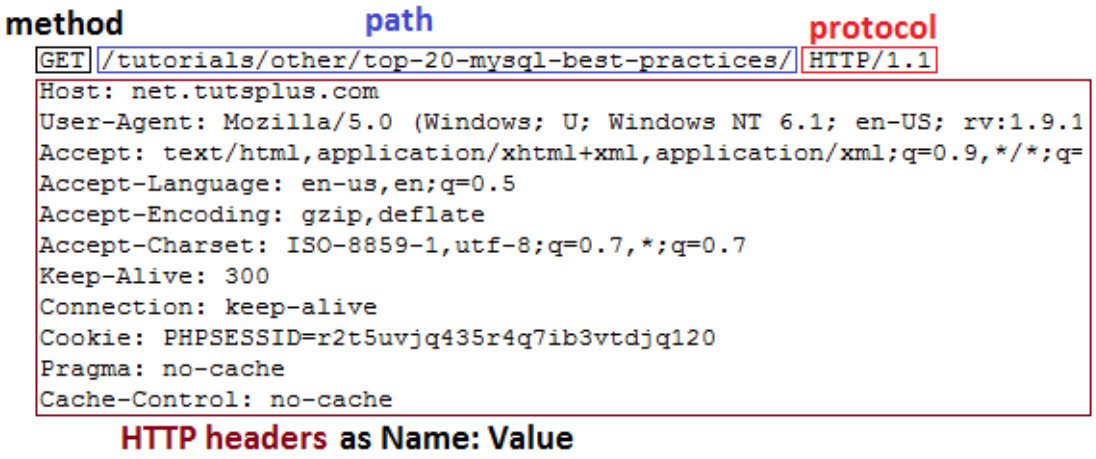
3.4 Github-Spider-Proxy-Rotated

Github-Spider-Proxy-Rotated is an advanced version of the previous spider; it contains pipelines, middlewares, and throttling parameters, which is in the middleware and settings file. This spider is for advanced scraping. It has a bottleneck mechanism for limiting the number of HTTP requests per second, and also a technique for HTTP headers rotation. It outputs the results in JSON format.



3.5 User-Agent-Spider

This spider gets the latest user-agent headers from an online forum, output of this file is directly put into use in other two spiders namely Github-Spider and Github-Spider-Proxy-Rotated. The result of this spider is a text file containing a list of HTTP headers which can is useful for spoofing browser's HTTP request activity.



We are using 3833 different proxies in our code which is generated using our proxy-spider.

3.6 Wiki-Satck-KML-Downloader

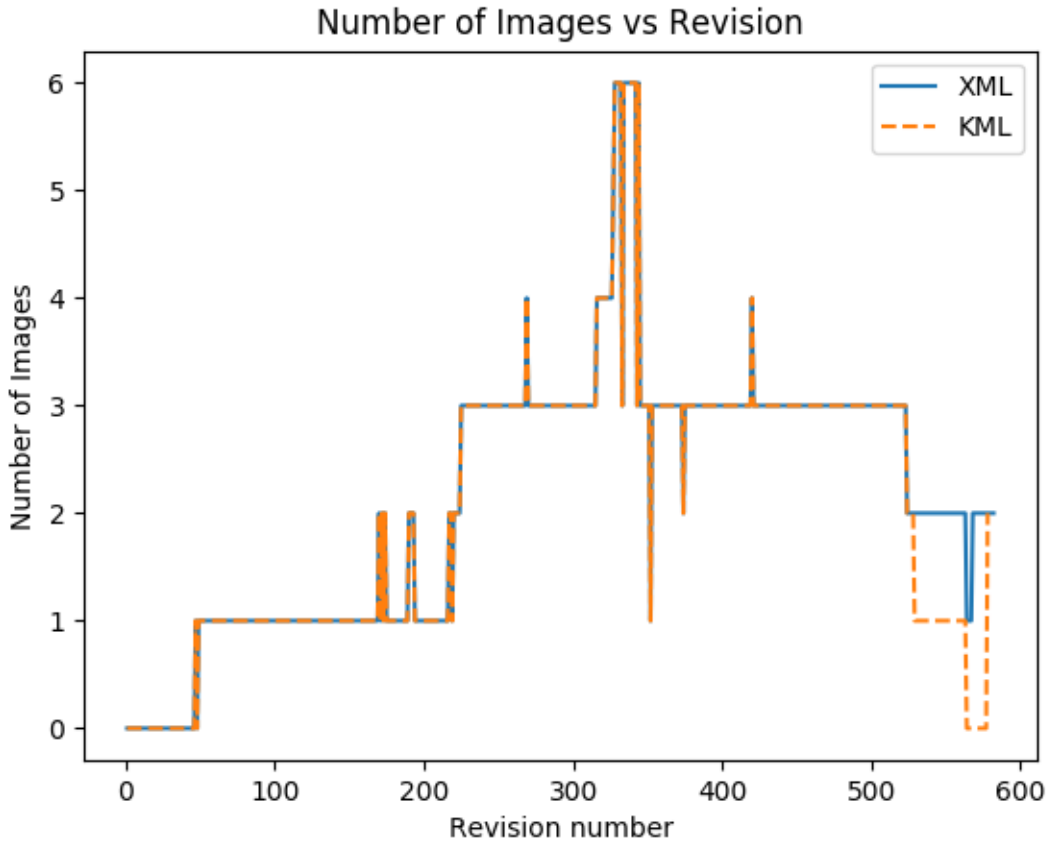
Dowloads xml from Wikipedia and Stackexchange for any given topic from user then coverts it to KML. The topics for wikipedia can be name of any page on wikipedia like UNO while for stackexchange the topics can any of the sub-site name of stackexchange like beer for beer.stackexchange.com, all these sites are listed in site_list.txt inside Wiki-Stack-KML-Downloader folder.

WIKIPEDIA
The Free Encyclopedia

4 Analysis

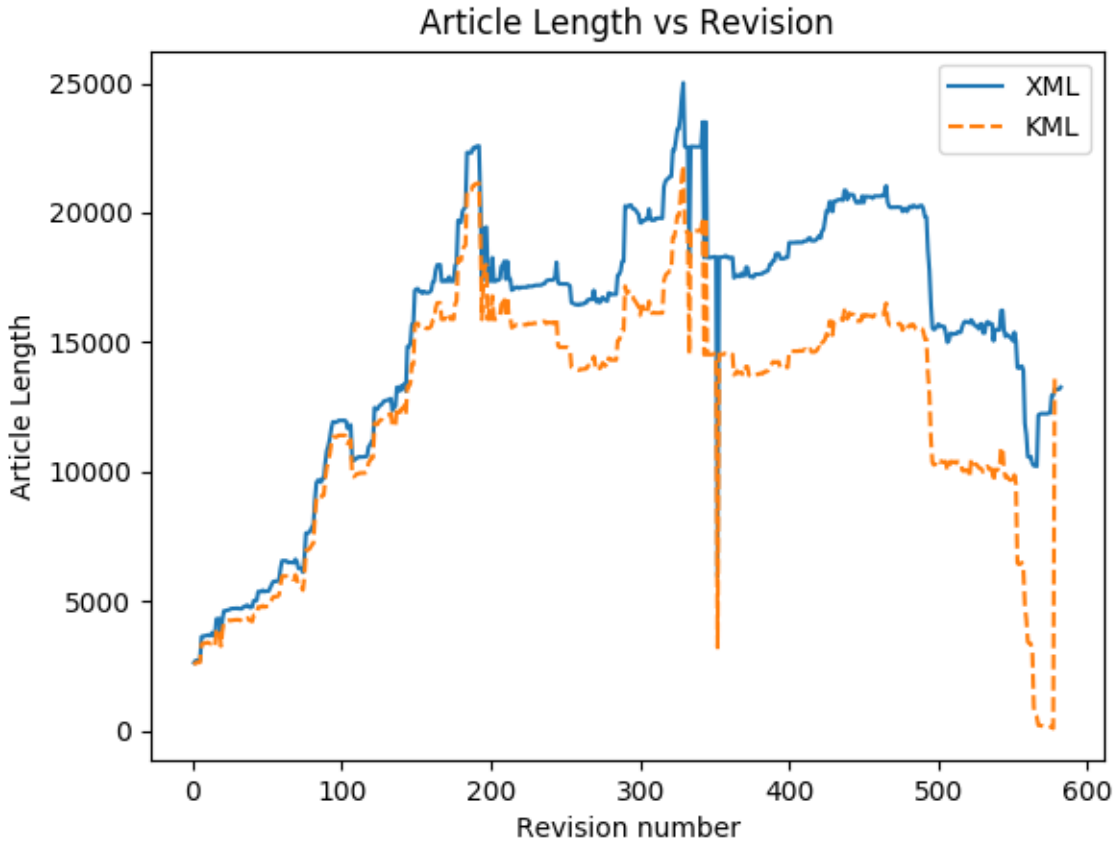
Graphs are generated to compare compressed KML and XML on various performance metrics like change in ArticleLength Vs Revision for KML and XML files which are plotted on same graph. Similarly, we have performed analysis based on change in Number of revisons, Number of Images, Number of Nouns (using natural language processing module **nltk**), Number of sections etc vs. Revision number and all these graphs are stored in AnalysisFig folder. Finally, we plotted the running time of these analysis functions for compressed XML and KML documnets and plotted on the same graph to show how a compressed KML can improve the runtime of various analysis functions with respect to a simple XML.

4.1 Number of Images vs. Revision



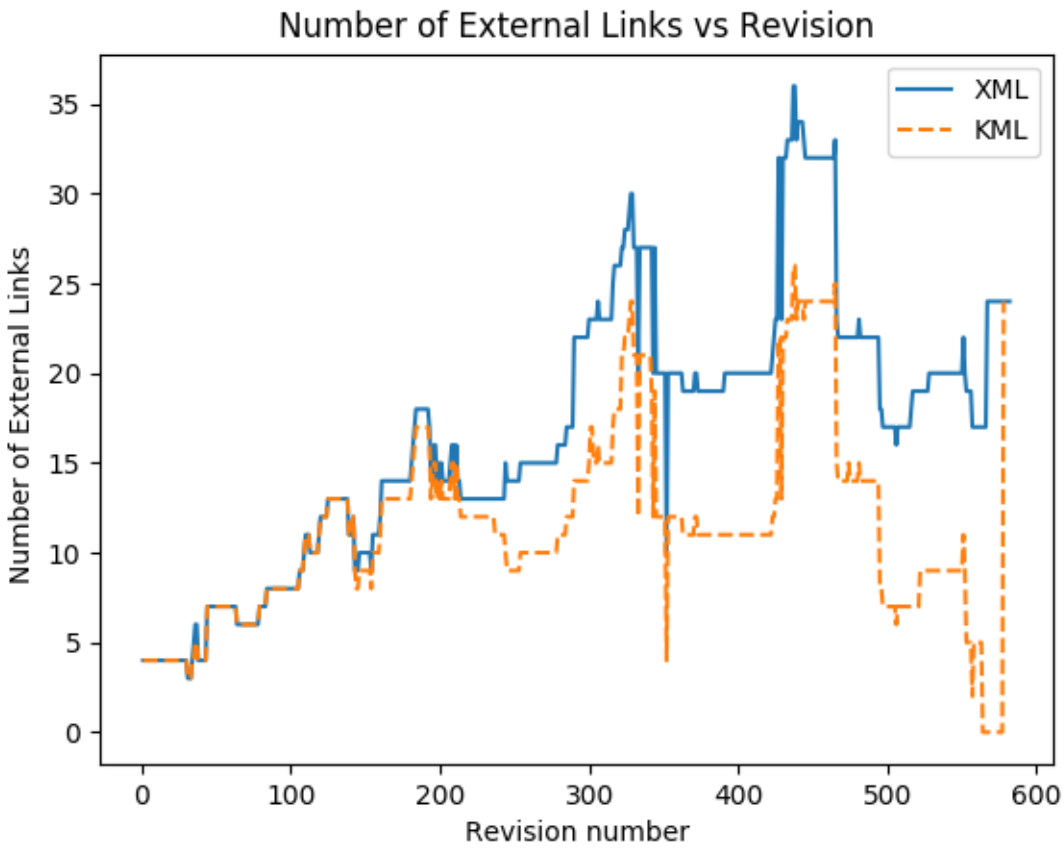
The compressed KML file stores all the images in the latest revision, so it has the same number of images as XML, but, other revisions store only those images which are different the latest revision, so they have a lesser number of images than XML.

4.2 Article Length vs. Revision



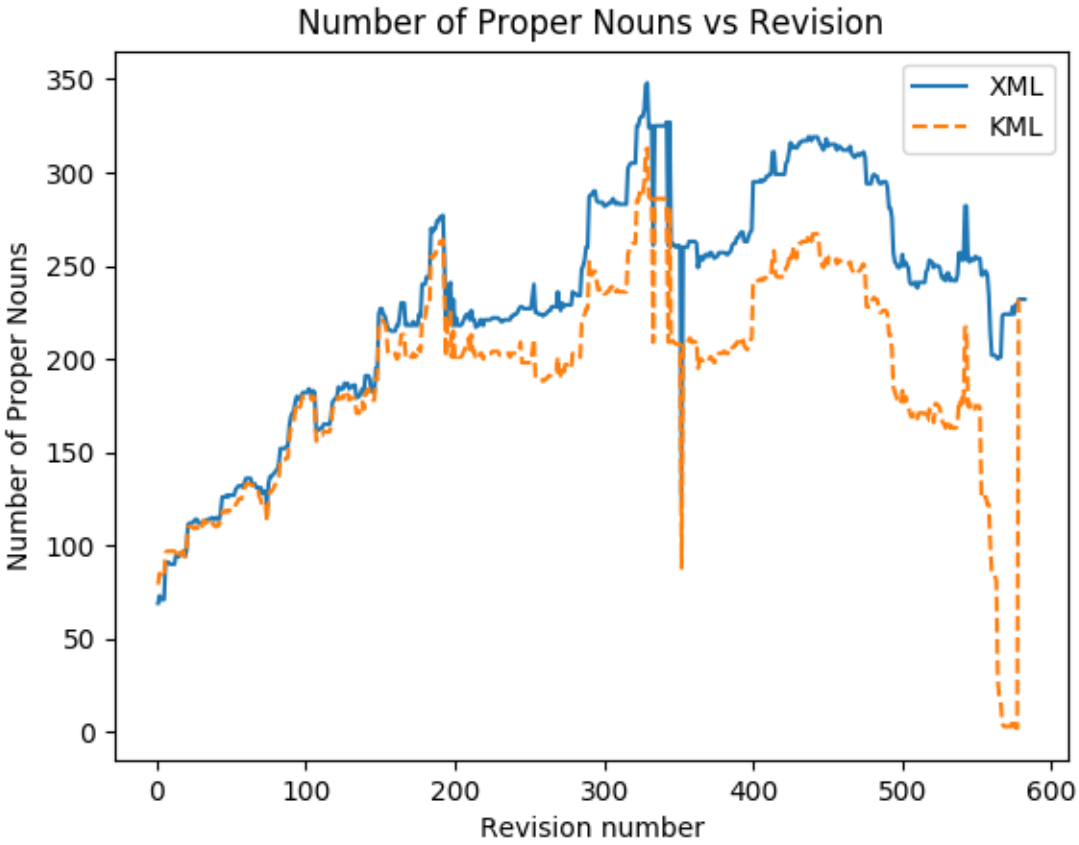
The compressed KML file stores the latest revision completely, so it has the same size as XML, but, other revisions store only the difference from the latest revision, so they take lesser space than XML.

4.3 Number of External Links vs. Revision



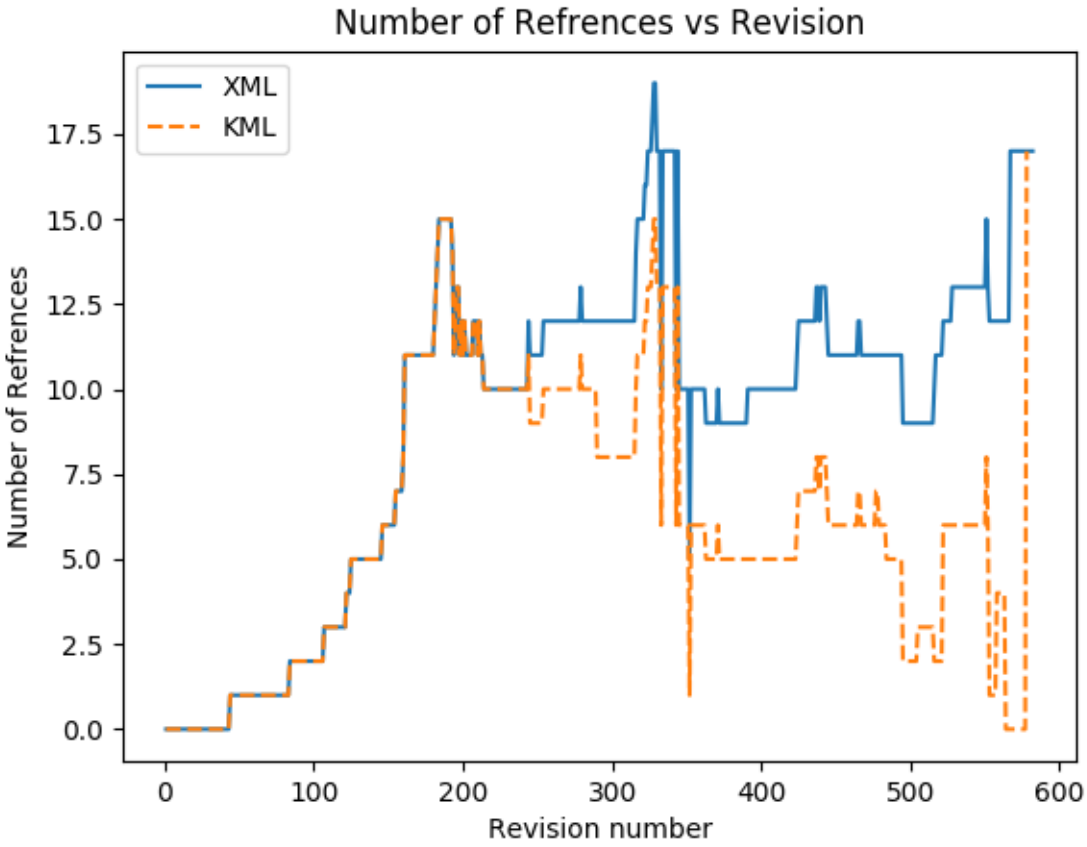
The compressed KML file stores all the links in the latest revision, so it has the same number of links as XML, but, other revisions store only the difference from the latest revision, so they have a lesser number of links than XML.

4.4 Number of Proper Nouns vs. Revision



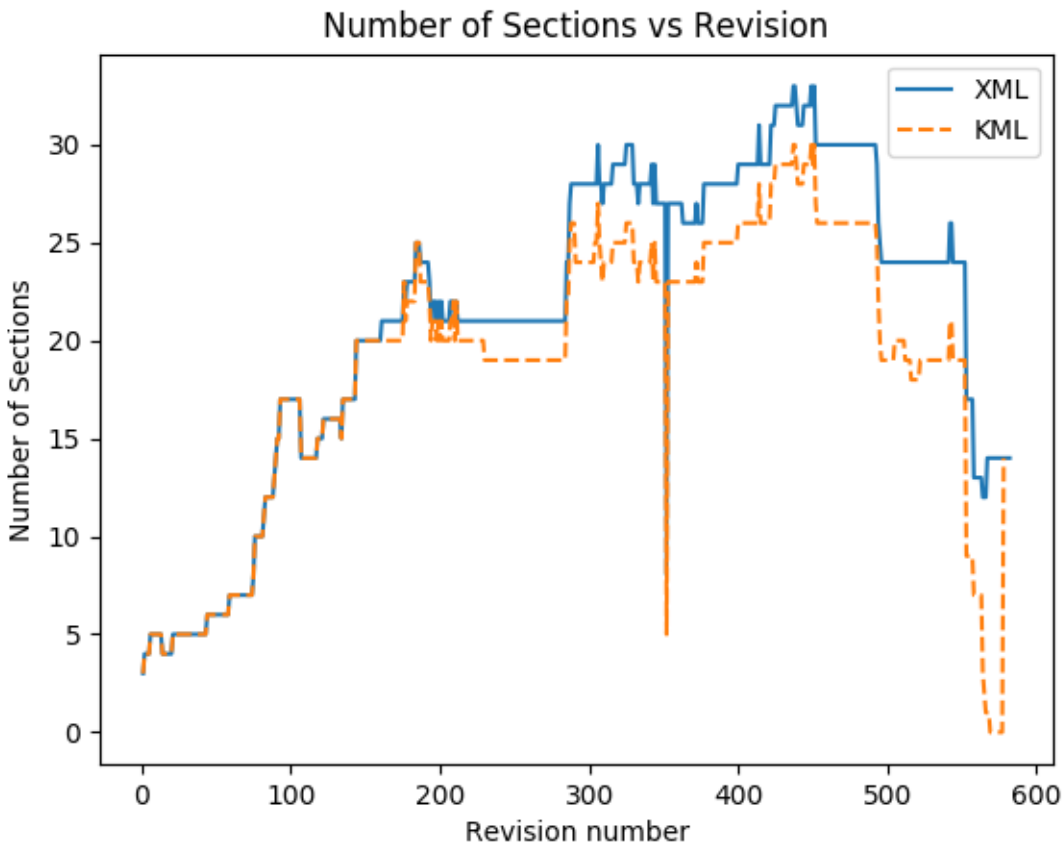
The compressed KML file has all the proper nouns in the latest revision, so it has the same number of proper nouns as XML, but, other revisions have only those proper nouns which are different from the latest revision, so they have a lesser number of proper nouns than XML.

4.5 Number of References vs. Revision



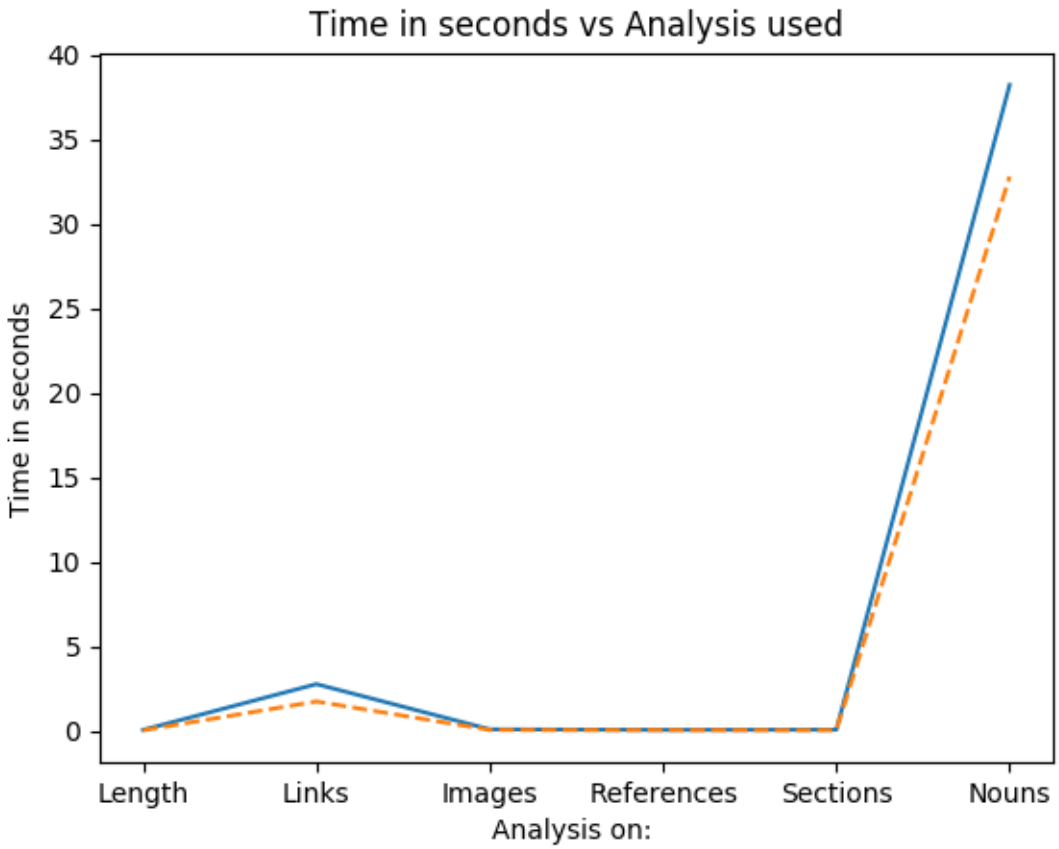
The compressed KML file stores all the references in the latest revision, so it has the same number of references as XML, but, other revisions store only the difference from the latest revision, so they have a lesser number of references than XML.

4.6 Number of Sections vs. Revision



The compressed KML file stores all the sections in the latest revision, so it has the same number of sections as XML, but, other revisions store only the difference from the latest revision, so they have a lesser number of sections than XML.

4.7 Time in seconds vs. Analysis used



As we already observed that each revision is smaller in compressed KML than a regular XML file, so all the analysis programs run faster on compressed KML. The speedup will be more when the analysis is complex like our Proper Noun vs Revision analysis where a speed of 5 seconds was observed.

5 Link to our Repository

URL: `https://github.com/csl-622/KML`