

Even Cheng - 41123717
Karthik Raman - 95687560
CS221-IR
March 18, 2010

INFORMATION RETRIEVAL ASSIGNMENT 6

PROTOTYPE OF A BILINGUAL SEARCH ENGINE

Suppose you are bilingual. Your search engine lets you search in one language at a time. But you want the best results for your search, regardless of language.

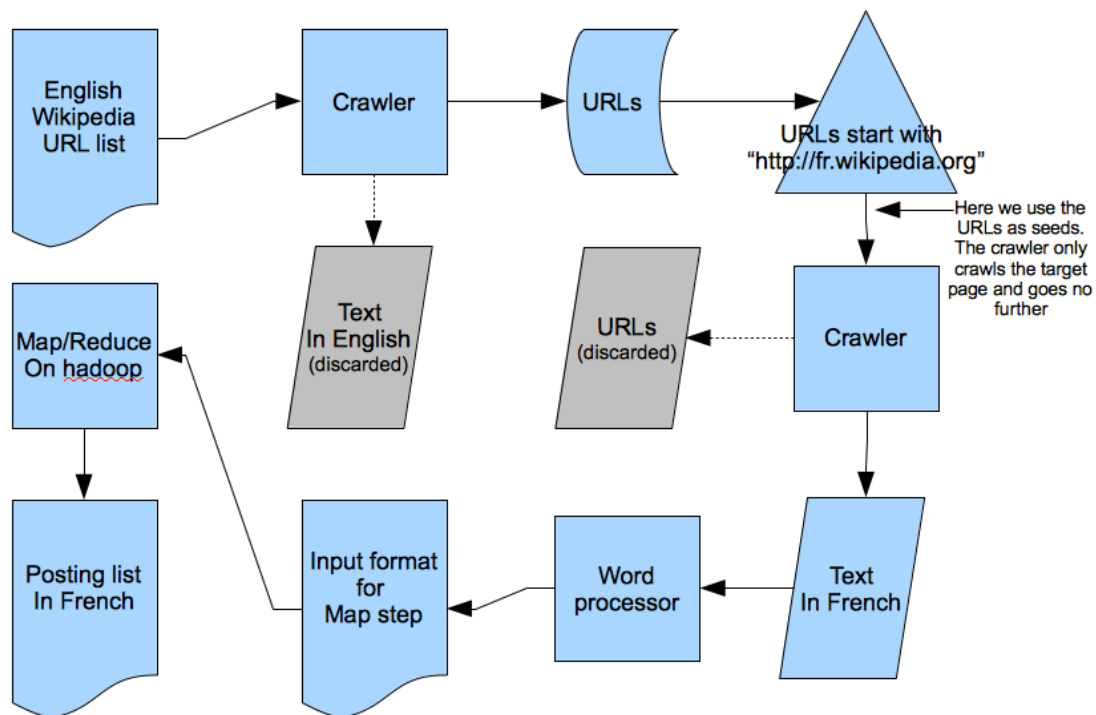
We have built a prototype of a bilingual search engine for Wikipedia that will search queries in both English and French and present the top-ranking results regardless of language.

Architecture

Our search engine consists of three major components: the **crawler**, the **query engine**, the **presentation UI**. The components are not dynamically interconnected but it is conceivable to build a system where they are.

Crawler

The following graph will illustrate the structure of crawler



We started from parsing the 500,000 URL list in English and crawl the English Wikipedia web pages back. The crawler4j help us parsing the web pages into plain text and a string list of URL In the crawler. We create a loop to search for the link in the same topic in French, which usually locates at the left-bottom part of the page. Once we retrieved the link starts with "<http://fr.wikipedia.org/wiki/>", we store it into our URL list in French and also call the crawler function again to retrieve the web page in French. After successfully retrieved the data we need, the following procedure is much the same as what we have done in project 4. We remove characters not belong to neither 0-9a-zA-Z nor the French accent character and finally combine the docid with the word as our input data for hadoop. After Map/Reduce is done, we collected posting lists and save the URLs scraped along with their document IDs. We sanitized the posting lists according to the directions in Assignment 5. We removed accented letters from the French posting lists and replaced them with their ASCII analogues, viz.¹:

- à -> a
- â -> a
- ä -> a
- è -> e
- é -> e
- ê -> e
- ë -> e
- î -> i
- ï -> i
- ô -> o
- œ -> ae
- ù -> u
- û -> u
- ü -> u
- ŷ -> y
- ç -> c

We supply the posting lists and URL lists as input to a binary file creator (see **BinaryPostingListCreator.java** in the Appendix). This program takes a file and mode as input. If the mode is "url" then it uses the file pointed to by the first argument to create two random-access files for accessing URLs, urls.data and urls.offset. Otherwise it uses the file pointed to by the first argument to create two random-access files for accessing posting lists, plist.data and plist.offset. We believe having a random-access file for URLs is an important optimization too. We ran into an overflow error in processing large posting lists; Java's short datatype was unable to hold the size of such lists. We used the integer datatype instead to store the lengths of posting lists.

¹ <http://tlt.its.psu.edu/suggestions/international/bylanguage/french.html#winaltthis>

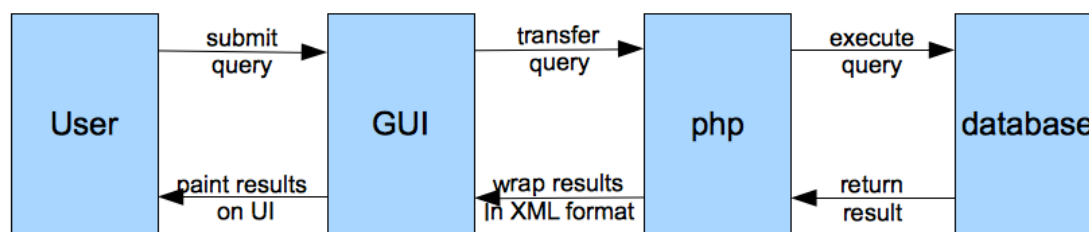
We supply the posting and URL binary files for English and French to the query engine. In addition, we supply an English–French dictionary file as input². This file has about 3000 entries. As with the French posting list, we sanitized it to remove accented letters. We use this dictionary file to create a one-to-many translations hash map between English and French and one-to-one hash map between French and English.

Query Engine

The query engine (**QueryEngine.java**) loads the input files from the previous steps and searches both corpora. We assume that the user will conduct searches in sessions, sessions being identified by the language of the input queries. It is possible, however, to make the language of each query variable, although we have not implemented this. In each session, the query engine searches the first corpus in the language of the query and the second corpus with the translated query. In the case of English–French translation, the query in French is simply the aggregate of the unique translation words from English.

User Interface

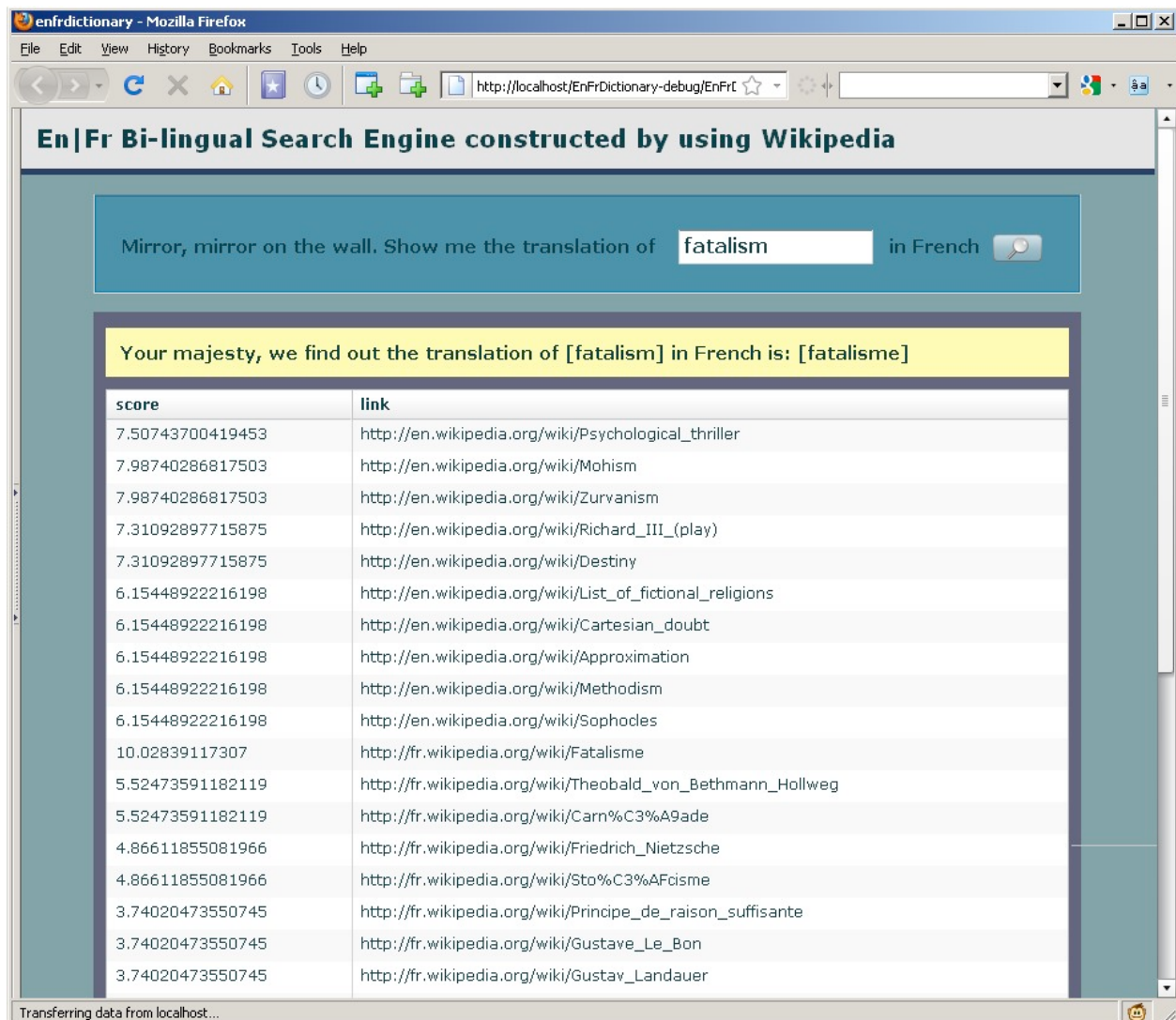
We designed a graphical user interface by Flex builder. The layout imitates search engines nowadays which run their search result in advance and store them in a database. As users their query, the query goes into database and return the prepared result instead of running the posting list and sorting repeatedly so that users can have their speedy search experience.



The structure of our user interface and its back end are illustrated as the graph above. After user submit a query, the user interface will pass the query to php and php will execute a query and wrap the results into XML format. Our UI will parse the XML format and paint the results on the screen. The following two figures are column structure in our database, and the web user interface.

² We used the dictionary file from <http://www.ilovelanguages.com/IDP/files/French.txt>

column name	type
id	int (auto_increment)
queryString	varchar(200)
result	varchar(200)
score	double
link	varchar(200)



Design Decisions

We use a variable mode to set up the language of a search session. In our prototype we have set the mode to be English by default, meaning the language

of search queries is English. The original query is searched in the English corpus and translations of the query searched in the French corpus. We could easily extend the system to accept English or French as the search mode.

An advantage of conducting search in sessions is efficiency. The most time-consuming piece of the query engine is the loading of the data files and data structures into memory. Once these are in memory, searching and sorting results is quick. Therefore, we implemented a loop to reset query scores to let the user query the engine indefinitely in a session to exploit the structures in memory.

Implementation

We implement searching and ranking using the cosine-similarity algorithm, ignoring normalization using document lengths. We used heap-search to extract the top 10 results in either language, and again to extract the top 10 results from that set of 20.

Parameters and Constraints

In assignment 5, we saw that Java could not handle large regexes on large strings that are in the range of 100s of KB. We had to restrict our search over posting lists less than 10000 bytes in length; this was even more restrictive than the 50,000 collection-frequency constraint. In addition, we observed that Java would run out of memory while creating a hash table with the entire posting list scraped in assignment 4. In this assignment we were able to overcome some, but not all, of these constraints.

In our development machine, we maxed out the configurable memory limit of the Java Virtual Machine (JVM); we ran our program with these settings:

```
java -Xms1536m -Xss32m
```

Here, the minimum size of JVM's memory is set to 1536Mb, minimum stack size set to 32m. With these settings we could load both corpora into memory:

- **Size of data structures on disk.** 1.92 GB for the two posting lists, two URL lists, and dictionary.
- **Size of data structures in memory.** 1.1 GB for the two posting lists, two URL lists, and dictionary. We store our posting list offset (1MB), and URL offset files (4MB) in memory. We saved approximately 8MB in not storing the full URL list in memory, although this is a miniscule saving.

Despite the availability of over 1GB of memory, we observed stack overflows on searches with English words that had many French translations. Therefore, we

were limited to words with limited number of translations. An implementation in a language without managed memory could circumvent this limitation of Java's.

Our searches are limited to words in the dictionary, because translated searches are not possible otherwise. A solution is keep growing the dictionary dynamically by asking user for possible translations if one is not found in the dictionary.

Results

In the query engine, the loading time of the corpora and other data structures takes the longest. After this, for each query, search time is in the order of 50ms. Sort time is in the order of 1000ms (10s).

Here are the results of three queries. Surprisingly, the top results were overwhelmingly in one language or the other. Results with a good mix of the results in two languages were rare. This suggests topic-related bias in language-specific searches. This is something to be explored in future work.

Query 1. Loading the query engine, querying “fatalism” in English

```
# java -Xms1536m -Xss32m QueryEngine ./ "fatalism" "EN"
fatalism      fatalisme
Search time is 10ms.
```

```
Top English hits:
Sort time is 7902ms
```

1. <http://en.wikipedia.org/wiki/Zurvanism>
2. <http://en.wikipedia.org/wiki/Mohism>
3. http://en.wikipedia.org/wiki/Psychological_thriller
4. [http://en.wikipedia.org/wiki/Richard_III_\(play\)](http://en.wikipedia.org/wiki/Richard_III_(play))
5. <http://en.wikipedia.org/wiki/Destiny>
6. http://en.wikipedia.org/wiki/List_of_fictional_religions
7. http://en.wikipedia.org/wiki/Cartesian_doubt
8. <http://en.wikipedia.org/wiki/Approximation>
9. <http://en.wikipedia.org/wiki/Methodism>
10. <http://en.wikipedia.org/wiki/Sophocles>

```
Top French hits:
Sort time is 9103ms.
```

1. <http://fr.wikipedia.org/wiki/Fatalisme>
2. http://fr.wikipedia.org/wiki/Theobald_von_Bethmann_Hollweg
3. <http://fr.wikipedia.org/wiki/Carn%C3%A9ade>
4. http://fr.wikipedia.org/wiki/Friedrich_Nietzsche
5. <http://fr.wikipedia.org/wiki/Sto%C3%AFcisme>
6. http://fr.wikipedia.org/wiki/Principe_de_raison_suffisante
7. http://fr.wikipedia.org/wiki/Gustave_Le_Bon
8. http://fr.wikipedia.org/wiki/Gustav_Landauer
9. <http://fr.wikipedia.org/wiki/Deuil>
10. http://fr.wikipedia.org/wiki/James_K._Galbraith

Top bilingual hits:

1. <http://fr.wikipedia.org/wiki/Fatalisme>
2. http://fr.wikipedia.org/wiki/Theobald_von_Bethmann_Hollweg
3. <http://fr.wikipedia.org/wiki/Carn%C3%A9ade>
4. http://fr.wikipedia.org/wiki/Friedrich_Nietzsche
5. <http://fr.wikipedia.org/wiki/Sto%C3%AFcisme>
6. http://fr.wikipedia.org/wiki/Principe_de_raison_suffisante
7. http://fr.wikipedia.org/wiki/Gustave_Le_Bon
8. http://fr.wikipedia.org/wiki/Gustav_Landauer
9. <http://fr.wikipedia.org/wiki/Deuil>
10. http://fr.wikipedia.org/wiki/James_K._Galbraith

Query 2. “yam” in English

Enter new query: yam
yam patate douce
Search time is 151ms.

Top English hits:

Sort time is 6890ms.

1. [http://en.wikipedia.org/wiki/Yam_\(vegetable\)](http://en.wikipedia.org/wiki/Yam_(vegetable))
2. [http://en.wikipedia.org/wiki/Yam_\(god\)](http://en.wikipedia.org/wiki/Yam_(god))
3. <http://en.wikipedia.org/wiki/Yam>
4. <http://en.wikipedia.org/wiki/Popeye>
5. http://en.wikipedia.org/wiki/Israel_national_ice_hockey_team
6. http://en.wikipedia.org/wiki/Cumbrian_dialect
7. http://en.wikipedia.org/wiki/Igbo_people
8. http://en.wikipedia.org/wiki/Mongol_Empire
9. http://en.wikipedia.org/wiki/Red_Sea
10. <http://en.wikipedia.org/wiki/Caesarea>

Top French hits:

Sort time is 10134ms.

1. http://fr.wikipedia.org/wiki/Patate_douce
2. http://fr.wikipedia.org/wiki/Pomme_de_terre
3. <http://fr.wikipedia.org/wiki/Igname>
4. <http://fr.wikipedia.org/wiki/Kisumu>
5. <http://fr.wikipedia.org/wiki/Nouvelle-Z%C3%A9lande>
6. <http://fr.wikipedia.org/wiki/Nouvelle-Z%C3%A9lande>
7. http://fr.wikipedia.org/wiki/Culture_de_la_R%C3%A9publique_du_Congo
8. <http://fr.wikipedia.org/wiki/Ta%C3%AFnos>
9. <http://fr.wikipedia.org/wiki/Mangaia>
10. <http://fr.wikipedia.org/wiki/Fruit>

Top bilingual hits:

1. http://fr.wikipedia.org/wiki/Patate_douce
2. http://fr.wikipedia.org/wiki/Pomme_de_terre
3. <http://fr.wikipedia.org/wiki/Igname>
4. <http://fr.wikipedia.org/wiki/Kisumu>
5. <http://fr.wikipedia.org/wiki/Nouvelle-Z%C3%A9lande>
6. <http://fr.wikipedia.org/wiki/Nouvelle-Z%C3%A9lande>
7. http://fr.wikipedia.org/wiki/Culture_de_la_R%C3%A9publique_du_Congo
8. <http://fr.wikipedia.org/wiki/Ta%C3%AFnos>
9. <http://fr.wikipedia.org/wiki/Mangaia>

10. <http://fr.wikipedia.org/wiki/Fruit>

Query 3. “abacus” in English

Enter new query: abacus
abacus boulier, compteur
Search time is 120ms.

Top English hits:

Sort time is 6830ms.

1. <http://en.wikipedia.org/wiki/Abacus>
2. http://en.wikipedia.org/wiki/Thomas_More
3. <http://en.wikipedia.org/wiki/Conscience>
4. [http://en.wikipedia.org/wiki/Division_\(mathematics\)](http://en.wikipedia.org/wiki/Division_(mathematics))
5. http://en.wikipedia.org/wiki/Roman_commerce
6. http://en.wikipedia.org/wiki/Italian_general_election,_2006
7. http://en.wikipedia.org/wiki/History_of_science_and_technology_in_China
8. <http://en.wikipedia.org/wiki/Calculator>
9. http://en.wikipedia.org/wiki/Stuyvesant_High_School
10. <http://en.wikipedia.org/wiki/Addition>

Top French hits:

Sort time is 9935ms.

1. <http://fr.wikipedia.org/wiki/Compteur>
2. http://fr.wikipedia.org/wiki/Techniques_de_calcul_mental
3. http://fr.wikipedia.org/wiki/Intel_8051
4. http://fr.wikipedia.org/wiki/Compteur_proportionnel_%C3%A0_gaz
5. http://fr.wikipedia.org/wiki/Compteur_de_bicyclette
6. http://fr.wikipedia.org/wiki/Liste_des_circuits_int%C3%A9gr%C3%A9s_de_la_s%C3%A9rie_7400
7. [http://fr.wikipedia.org/wiki/Fermeture_\(informatique\)](http://fr.wikipedia.org/wiki/Fermeture_(informatique))
8. http://fr.wikipedia.org/wiki/Rayon_X
9. http://fr.wikipedia.org/wiki/Fraude_%C3%A9lectorale
10. http://fr.wikipedia.org/wiki/Projet_Dharma

Top bilingual hits:

1. <http://en.wikipedia.org/wiki/Abacus>
2. http://en.wikipedia.org/wiki/Thomas_More
3. <http://en.wikipedia.org/wiki/Conscience>
4. [http://en.wikipedia.org/wiki/Division_\(mathematics\)](http://en.wikipedia.org/wiki/Division_(mathematics))
5. http://en.wikipedia.org/wiki/Roman_commerce
6. http://en.wikipedia.org/wiki/Italian_general_election,_2006
7. http://en.wikipedia.org/wiki/History_of_science_and_technology_in_China
8. <http://en.wikipedia.org/wiki/Calculator>
9. http://en.wikipedia.org/wiki/Stuyvesant_High_School
10. <http://en.wikipedia.org/wiki/Addition>

Appendix: Source Code Listing

FrenchHadoop.java

```
package edu.uci.ics.ir.hw4;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
```

```

import java.util.Set;
import java.util.StringTokenizer;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

import edu.uci.ics.crawler4j.crawler.HTMLParser;
import edu.uci.ics.crawler4j.crawler.Page;
import edu.uci.ics.crawler4j.crawler.PageFetchStatus;
import edu.uci.ics.crawler4j.crawler.PageFetcher;
import edu.uci.ics.crawler4j.url.WebURL;

import org.apache.hadoop.fs.*;
import org.apache.hadoop.mapred.JobConf;

public class HadoopHeavy {

    private HTMLParser htmlParser = new HTMLParser();

    // for download data from wikipedia, use here
    private static int startToFetchNumber = 0;
    private static int endToFetchNumber = 500000;
    private static final int documentsPerOutput = 10000;

    WebURL curURL;
    Page page;
    private static BufferedWriter URLWriter;
    public Page download(String url, int docid) {
        curURL = new WebURL(url, docid);
        page = new Page(curURL);
        page.setWebURL(curURL);
        PageFetcher.setPolitenessDelay(0);
        return FetchPage(page, docid);
    }

    public Page FetchPage(Page page, int docid){
        Set<String> linkSet;
        int statusCode = PageFetcher.fetch(page);
        if (statusCode == PageFetchStatus.OK) {
            try {
                htmlParser.parse(page.getHTML(), page.getWebURL().getURL());
                linkSet = htmlParser.getLinks();
                for(String link : linkSet){
                    if(link.startsWith("http://fr.wikipedia.org/wiki/")){
                        curURL = new WebURL(link, docid);
                        page = new Page(curURL);
                        page.setWebURL(curURL);

                        URLWriter.write(docid + "\t" + curURL + "\n");
                        // crawl again
                        statusCode = PageFetcher.fetch(page);
                        if (statusCode == PageFetchStatus.OK) {
                            htmlParser.parse(page.getHTML(),
                                page.getWebURL().getURL());

                            page.setText(htmlParser.getText());
                            page.setTitle(htmlParser.getTitle());
                            return page;
                        }
                    }
                }
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
}

```

```

        return null;
    }

    public static void main(String[] args) throws Exception {
        HadoopHeavy myHadoop = new HadoopHeavy();
        if(args.length < 3){
            System.out.println("Usage: java -jar HadoopHeavy.jar [inputLinks]
[inputFolder] [outputFolder] [startpage] [endpage] [URLWriterPath]");
            return;
        }

        JobConf conf = new JobConf(HadoopHeavy.class);
        if(args.length > 5)            URLWriter = new BufferedWriter(new FileWriter(args
[5]));

        System.out.println(Path.CUR_DIR);
        Path linkpath = new Path(Path.CUR_DIR, "/" + args[0]);
        Path outpath = new Path(Path.CUR_DIR, "/" + args[2] );
        FileSystem inputLinks = linkpath.getFileSystem(conf);
        System.out.println("Delete output directory " + args[2] + ": " + inputLinks.delete
(outpath, true));

        if(args.length > 3){
            startToFetchNumber = Integer.parseInt(args[3]);
            endToFetchNumber = Integer.parseInt(args[4]);
        }

        try{
            FSDataInputStream links = inputLinks.open(linkpath);
            String tmp = "";
            //Pattern p = Pattern.compile("[^a-zA-Z0-9 ]");
            Pattern p;// = Pattern.compile("[^a-zA-Z0-9]+[ \\t\\n\\x0B\\f\\r]+");
            Matcher m;
            Path inpath;
            int docid;
            int i = - 1;

            // write on dfs system
            FSDataOutputStream wordcount = null;

            // write locally
            BufferedWriter bw = null;

            while((tmp = links.readLine()) != null){
                i++;
                if(i % documentsPerOutput == 0){
                    System.out.println(i/10000);
                }
                if(i < startToFetchNumber)        continue;
                if(i > endToFetchNumber){
                    bw.flush();
                    bw.close();
                    break;
                }
                if(i % documentsPerOutput == 0){
                    if(bw != null){
                        bw.flush();
                        bw.close();
                    }
                    bw = new BufferedWriter(new FileWriter(args[1] + "/" + i /
documentsPerOutput + "output.txt"));
                }

                if(wordcount != null){
                    wordcount.flush();
                    wordcount.close();
                }
            }
        }
    }
}

```

```

    }
    inpath = new Path(Path.CUR_DIR, "/" + args[1] + "/" + i /
documentsPerOutput + "output.txt");
    wordcount = inputLinks.create(inpath);
}

String param[] = tmp.split("\t");
docid = Integer.parseInt(param[1]);
Page page = myHadoop.download("http://en.wikipedia.org/wiki/" +
param[0], docid);

if (page != null) {
    String fullpage = page.getText();

    p = Pattern.compile("[^000-177]&[^192-214]&[^216-246]
&[^248-255]");
    m = p.matcher(fullpage);
    while(m.find()){
        fullpage = m.replaceAll(" ");
        m = p.matcher(fullpage);
    }

    p = Pattern.compile("(\\p{Punct})|([ \\t\\n\\x0B\\f\\r]+
[^a-zA-Z0-9]+)|([a-zA-Z0-9]+[ \\t\\n\\x0B\\f\\r]+)");
    m = p.matcher(fullpage);
    while(m.find()){
        fullpage = m.replaceAll(" ");
        m = p.matcher(fullpage);
    }

    fullpage = fullpage.toLowerCase();
    String[] termArr = fullpage.split("[ \\t\\n\\x0B\\f\\r]+");
    fullpage = "";
    for(int j = 0; j< termArr.length;j++) {
        if (termArr[j].length()>32){
            termArr[j]= termArr[j].substring(0,31);
        }
        wordcount.writeChars(termArr[j] + " " + docid + "
");

        bw.write(termArr[j] + " " + docid + " ");
    }
    }
    wordcount.writeChars("\n");
    bw.write("\n");
}
wordcount.flush();
wordcount.close();
URLWriter.flush();
URLWriter.close();
bw.flush();
bw.close();
}
catch(IOException e){
    e.printStackTrace();
}

PostingList p = new PostingList();

String[] newargs = {args[1], args[2]};
try{
    p.run(newargs);
}
catch(Exception e){
    e.printStackTrace();
}

```

```

    }
}

```

PostingList.java

```

package edu.uci.ics.ir.hw4;

import java.io.IOException;
import java.util.Collections;
import java.util.Iterator;
import java.util.StringTokenizer;
import java.util.Vector;
import java.util.HashMap;
import org.apache.hadoop.filecache.DistributedCache;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.FileInputFormat;
import org.apache.hadoop.mapred.FileOutputFormat;
import org.apache.hadoop.mapred.JobClient;
import org.apache.hadoop.mapred.JobConf;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.Mapper;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reducer;
import org.apache.hadoop.mapred.Reporter;
import org.apache.hadoop.mapred.TextInputFormat;
import org.apache.hadoop.mapred.TextOutputFormat;

@SuppressWarnings("deprecation")
public class PostingList {
    public static class Map extends MapReduceBase implements
        Mapper<LongWritable, Text, Text, IntWritable> {
        private Text word = new Text();
        IntWritable docid = new IntWritable(0);
        public void map(LongWritable key, Text value,
            OutputCollector<Text, IntWritable> output, Reporter reporter)throws IOException {
            StringTokenizer itr = new StringTokenizer(value.toString());
            while (itr.hasMoreTokens()) {
                try{
                    word.set(itr.nextToken());
                    if(itr.hasMoreTokens()){
                        docid.set(Integer.parseInt(itr.nextToken()));
                        output.collect(word, docid);
                    }else{
                    }
                }catch(Exception e){
                    System.out.println(e.getStackTrace());
                }
            }
        }
    }

    public static class Reduce extends MapReduceBase implements
        Reducer<Text, IntWritable, Text, Text> {
        public void reduce(Text key, Iterator<IntWritable> values,
            OutputCollector<Text, Text> output, Reporter reporter)throws IOException {
            Vector<Integer> df = new Vector<Integer>();
            Integer docf = new Integer(0);
            Integer cf = new Integer(0);
            HashMap<Integer, Integer> tf = new HashMap<Integer, Integer>(200); // Change the
size of the HashMap according to how many docids
            while (values.hasNext()) {

```

```

        // Update collection frequency
        cf++;
        // Update document frequency
        int tmp = values.next().get();
        if(!df.contains(tmp)) { // in df, if docid is unseen, add it into df
            df.add(tmp);
        }
        // Create a new entry in HashMap for hitherto unseen docid
        tf.put(tmp,1);
        docf++;
    }
    // If document was seen, update its tf
    else {
        Integer curTF = (Integer) tf.get(tmp);
        ++curTF;
        tf.put(tmp, curTF);
    }
}

Vector<Integer> sortedset = new Vector<Integer>(tf.keySet());
Collections.sort(sortedset);
Iterator<Integer> itr = sortedset.iterator();
int docid = 0;
Vector<String> list = new Vector<String>();
while(itr.hasNext()){
    docid = itr.next();
    list.add(docid + "- " + tf.get(docid));
}
// Modify this to output [term, cf, df] [docid1, tf1] [docid2, tf2]

Text text = new Text();
text.set(cf + ", " + docf + "\t" + list.toString().replaceAll(", ", " ")
[ ]).replaceAll("- ", ", ");
output.collect(key, text);
}
}

public int run(String[] args) throws Exception {
    JobConf conf = new JobConf(PostingList.class);
    conf.setNumReduceTasks(1);
    DistributedCache.addFileToClassPath(new Path("/assnlibs/commons-codec-1.4.jar"),
conf);
    DistributedCache.addFileToClassPath(new Path("/assnlibs/commons-
logging-1.1.1.jar"), conf);
    DistributedCache.addFileToClassPath(new Path("/assnlibs/crawler4j-1.5.jar"),
conf);
    DistributedCache.addFileToClassPath(new Path("/assnlibs/dsiutils-1.0.10.jar"),
conf);
    DistributedCache.addFileToClassPath(new Path("/assnlibs/fastutil-5.1.5.jar"),
conf);
    DistributedCache.addFileToClassPath(new Path("/assnlibs/httpclient-4.0.1.jar"),
conf);
    DistributedCache.addFileToClassPath(new Path("/assnlibs/httpcore-4.0.1.jar"),
conf);
    DistributedCache.addFileToClassPath(new Path("/assnlibs/je-4.0.71.jar"), conf);
    DistributedCache.addFileToClassPath(new Path("/assnlibs/log4j-1.2.15.jar"), conf);

    conf.setJobName("PostingList");

    conf.setOutputKeyClass(Text.class);
    conf.setOutputValueClass(Text.class);

    conf.setMapOutputKeyClass(Text.class);
    conf.setMapOutputValueClass(IntWritable.class);
}

```

```

        conf.setMapperClass(Map.class);
        conf.setReducerClass(Reduce.class);

        conf.setInputFormat(TextInputFormat.class);
        conf.setOutputFormat(TextOutputFormat.class);

        FileInputFormat.setInputPaths(conf, new Path(args[0]));
        FileOutputFormat.setOutputPath(conf, new Path(args[1]));

        JobClient.runJob(conf);
        return 0;
    }
}

```

MXML file for user interface: **EnFrDictionary.mxml**

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute"
pageTitle="enfrdictionary"
    creationComplete="initApp()" backgroundColors="#ffffff, #ffffff">
    <mx:Script source="enfrdictionaryScript.as" />
    <mx:ViewStack id="applicationScreens" width="100%" height="100%">
        <mx:Canvas id="view" width="100%" height="100%" backgroundColor="#013F46"
backgroundAlpha="0.56">
            <mx:Canvas y="49" height="5" backgroundColor="#213151" right="18" left="0">
            </mx:Canvas>
            <mx:Canvas y="-1" height="50" backgroundColor="#DFDFDF" backgroundAlpha="1.0" left="0"
right="18">
                <mx:Label x="10" y="10" text="En|Fr Bi-lingual Search Engine constructed by using
Wikipedia" fontSize="20" fontWeight="bold"/>
            </mx:Canvas>
            <mx:VBox y="54" height="100%" horizontalAlign="center" left="0" right="18">
                <mx:Spacer height="10"/>
                <mx:Canvas width="807" height="81" backgroundColor="#00699F" backgroundAlpha="0.45"
cornerRadius="20" alpha="1.0" borderStyle="inset">
                    <mx:Label text="Mirror, mirror on the wall. Show me the translation of"
width="451" height="23" fontSize="15" x="19" y="29" fontWeight="bold"/>
                    <mx:Label text="in French" width="79" height="23" fontSize="15" x="646" y="29"
fontWeight="bold"/>
                    <mx:TextInput id="filterTxt" width="159" tooltip="Search by queryString"
enter="filterResults()" x="476" y="27" fontSize="16" fontWeight="bold"/>
                    <mx:Button click="filterResults()" id="filterButton0" icon="@Embed('icons/
SearchRecord.png')" tooltip="Search by queryString" x="733" y="31"/>
                </mx:Canvas>
                <mx:Spacer height="3"/>
                <mx:Canvas width="807" backgroundColor="#2A0036" backgroundAlpha="0.44">
                    <mx:VBox x="0" y="0" width="100%" horizontalAlign="center" verticalGap="10">
                        <mx:Spacer height="3"/>
                        <mx:Canvas height="40" backgroundColor="#FFCA7" width="787">
                            <mx:Label x="10" y="9" fontSize="15" fontWeight="bold"
id="translationString" fontStyle="normal">
                                </mx:Label>
                            </mx:Canvas>
                            <mx:DataGrid id="dataGrid"
                                dataProvider="{dataArr}"
                                rowCount="8"
                                editable="false"
                                headerRelease="setOrder(event);"
                                fontSize="12" minHeight="268" width="787" height="232"
                                resizableColumns="false" rowHeight="26">
                                <mx:columns>

```

```

        <mx:DataGridColumn headerText="score" width="200"
dataField="scoreCol" />
        <mx:DataGridColumn headerText="link" dataField="linkCol" />
    </mx:columns>
</mx:DataGrid>
<mx:Spacer height="3"/>
</mx:VBox>
</mx:Canvas>
<mx:Canvas height="72" width="100%">
    <mx:HRule y="10" height="3" right="0" left="0"/>
    <mx:Text y="21" text="Creators:    Even Cheng (huangyac at ics dot uci dot edu)
&#xa;Karthik Raman (ramankmail at gmail dot com)" textAlign="right" enabled="true"
selectable="true" right="10"/>
</mx:Canvas>
</mx:VBox>

</mx:Canvas>

<mx:Canvas id="update" width="100%" height="100%">
    <mx:Form width="100%" height="80%" id="enfrdictionForm">
        <mx:FormItem label="EnglishString:" id="EnglishString_form">
            <mx:TextInput id="EnglishStringCol" text=""/>
        </mx:FormItem>
        <mx:FormItem label="FrenchString:" id="FrenchString_form">
            <mx:TextInput id="FrenchStringCol" text=""/>
        </mx:FormItem>
        <mx:FormItem label="Score:" id="score_form">
            <mx:TextInput id="scoreCol" text=""/>
        </mx:FormItem>
        <mx:FormItem label="Link:" id="link_form">
            <mx:TextInput id="linkCol" text=""/>
        </mx:FormItem>
    </mx:Form>
    <mx:Button label="Save" id="btnSubmit" click="insertItem()" right="81" bottom="10"/>
    <mx:Button label="Cancel" id="btnCancel" click="goToView()" right="10" bottom="10"/>
</mx:Canvas>

</mx:ViewStack>
</mx:Application>

```