# Google Summer of Code, 2014
# Progress Report

**Organization:** DBpedia
**Project Name:** Abbreviation Base – A multilingual knowledge base for abbreviations
**Mentor Name**: Martin Brümmer

| Week | Implementation | Progress from previous week | Problems |
|---|---|---|---|
| Week 1 (April 28 - May 4) | Implemented sample lemon model for 4 abbreviations (from abbrevs_en.tsv in the github repository) in RDF/XML format with the following specification:<br>● Every language has its own lemon:Lexicon<br>● In every Lexicon, there are may lemon:LexicalEntry resources for the abbreviations.<br>● Every LexicalEntry has a lemon:LexicalForm. This will contain the abbreviation string.<br>● The LexicalEntry also has many lemon:LexicalSense resources. These contain the meaning of the abbreviations as a string and the link to the DBpedia page of the abbreviation as reference.<br>● The LexicalEntry also has a translation link that links the LexicalSense resource to its equivalent in English, if it exists.<br><br>The above model was validated using 'rapper-utils' and was found syntactically correct.<br><br>The correctness of a lemon file was tested with lemon validator: https://github.com/jmccrae/lemon-model.net/tree/master/validator and was found completely valid. | | Had some namespace problems which were resolved |
| | Similar implementation of abbreviation (4 abbreviations) was also done in turtle format with the above specification. | | The correctness of a lemon file was tested with lemon |

| | | | |
|---|---|---|---|
| | The above model was validated using 'rapper-utils' and was found syntactically correct.<br><br>[the .ttl file has been pushed to the github repository: https://github.com/der-bruemmer/abbreviation-base] | | validator: https://github.com/jmccrae/lemon-model.net/tree/master/validator and following error was found:<br><br>[ERROR] Failed to parse as RDF: file:///home/div/Desktop/abbreviation_lemon_model.ttl:1:0: not well-formed (invalid token) |
| Week 2 (May 5 - May 11 ) | Made a python script.<br><br>It did following things:<br>1. Downloaded the required files from DBpedia dump data for a language ('en' was tested), read them and saved them in .bz2 format.<br><br>2. Decompressed the zipped files saved in hard disk and then according to the regular expression, writes a .ttl file containing the triples ending with ., ? or ! (as required). I used the bz2 function BZ2File(file_name,'rb'), to decompress the files. | Made a python script which reads necessary files from DBpedia server and decompresses them and store them in hard disk. | The file abbrev_en.ttl in the repo had many triples that were not present in **http://downloads.dbpedia.org/3.9/en/redirects_en.ttl.bz2** . When comparing the file I have written via code, I found that triples after "<http://dbpedia.org/resource/American_Telephone_and_Telegraph,_Inc.> <http://dbpedia.org/ontology/wikiPageRedirects> <http://dbpedia.org/resource/AT&T_Corporation> ." are not present the redirects file that I had decompressed. |
| Week 3 (May 12 - May 19) | Improved the above code to automate the downloading and decompressing of files.<br><br>I executed the code for French (fr), English (en) and also made a time file with the total time taken by each file to download and decompress. | I used **os.popen** library of python to decompress the file. The problem of incomplete file reading was solved. | |

| | | | |
|---|---|---|---|
| | The script downloads 7 files namely:<br>1. redirects- contains abbreviations<br>2. instance_types_heuristic<br>3. interlanguage_links<br>4. interlanguage_links_chapters<br>5. labels<br>6. article_categories<br>7. disambiguations<br><br>All the things were automated. | Now all the files were downloaded successfully and decompressed completely.<br><br>The problem was due to a bug in **bz2** python library due to which I was unable to read complete file. | |
| **First Phase Completed** | ● Made the code to automate the downloading and decompressing of necessary files from DBpedia data dump.<br><br>● I executed the code for French (fr), English (en) and also made a time file with the total time taken by each file to download and decompress.<br><br>● The script downloads 7 files namely:<br>1. redirects- contains abbreviations<br>2. instance_types_heuristic<br>3. interlanguage_links<br>4. interlanguage_links_chapters<br>5. labels<br>6. article_categories<br>7. disambiguations<br><br>All the things were automated. | | |

| | | | |
|---|---|---|---|
| Week 4 (May 20 - May 26) | Importing the necessary files into local virtuoso.<br><br>(Setting up files into virtuoso so that it could be accessed using SPARQL queries)<br><br>Setting up isql console to do the same. | | I always get following error on executing: isql -v 1111 dba dba "[IM002][unixODBC][Driver Manager]Data source name not found, and no default driver specified [ISQL]ERROR: Could not SQLConnect"<br><br>Whenever I upload a big file around 2 GB in quad store upload in Virtuoso, I am automatically logged off and then I have to login again. It is not happening when I am uploading a smaller file. |

| | | | |
|---|---|---|---|
| Week 5 (May 27 - June 2) | Importing the necessary files into local virtuoso .<br><br>Upload the file via the isql-vt console and execute queries on it. | Virtuoso may not use the standard isql command but either isql-vt or isql-v or isql-t.<br><br>So I tried isql-vt and the isql front-end opened.<br><br>Files were uploaded successfully as I zipped big files which took a lot of time (as VOS can import files in gzip format). | The code took about 3 hours to run and import files. |
| Week 6 (June 3 - June 9) | Working with the extractor code, extending the code already present in repository, heavily changing the SPARQL queries to extract the required information and improving the code to store the extracted information for the abbreviation. | Files were imported successfully.<br><br>Tested SPARQL queries over uploaded data, the query is working perfectly fine for the abbreviation data. | Type and sameAs has multiple values for one abbreviation, so how is it to be fetched and how is to be stored and how will they be stored in Lemon file ?<br><br>Eg: select * where {<http://dbpedia.org/resource/BC_Card> rdf:type ?type }<br>(sense of <http://dbpedia.org/resource/BC> abbreviation B.C.) has following rdf:type<br><br>type http://www.w3.org/2002/07/owl#Thing http://schema.org/Organization http://dbpedia.org/ontology/Agent http://dbpedia.org/ontology/Company http:// |

| | | | |
|---|---|---|---|
| | | | dbpedia.org/ontology/Organisation<br><br>Solution as per me:<br>I will use a list to store the multiple values (type or sameAs) and will append in the existing dictionary in the code which I have already made.<br><br>Problem in fetching category and rdf:type of the abbreviations which do not have disambiguations. |
| Week 7 (June 9 - June 15) | Fetching the necessary details of abbreviations like label, type, sameAs, categories.<br><br>Multiple values like owl:sameAs, rdf:type, are stored in a list and stored it in the final list i.e. list in list.<br>The format is :<br><br>->for disambiguations:<br>**abbrevs[abbrev+" "+str(count)] = [result["label"]["value"], result["o"]["value"], list_sameAs, list_type]**<br><br>->for non-disambiguations:<br>**data = [result["label"]["value"], uri, list_sameAs]**<br><br>Data Fetched:<br>#----result["name"]["value"] ==> lemon:value "Michel van der Aa"@en<br>#----result["label"]["value"] ==> rdfs:label "Michel van der Aa<br>#----result["o"]["value"] ==> lemon:reference <http://dbpedia.org/resource/Michel_van_der_Aa> | Extended the code, made some queries to extract the necessary information related to the abbreviations like rdf:type, owl:sameAs, category and label.<br><br>Used list to store multiple values and stored this list into a final list making a 2D list.<br><br>Solution to multiple sameAs and type value by mentor:<br>We don't really need owl#Thing, but we want "dbpedia.org" or "schema.org") excludes any "yago" types, that are somewhat messy.<br><br>I have used REGEX in the query which | |

| | | | |
|---|---|---|---|
| | #----list_sameAs ==> owl:sameAs<br>#----list_type ==> rdf:type "but works only for abbreviations with disambiguations"<br>#----abbrevString ==> writtenRep , stores abbrev<br><br>The code is ready, it fetches the required information and stores it in a .tsv file.<br>I have tested the code for first five inputs and it works perfectly fine for both abbreviation with disambiguation and without it.<br>(categories and type will be fetched later.) | will only fetch the required data i.e. dbpedia and schema.<br><br>Solution to fetch category:<br>The category file being uploaded was incorrect.<br>The correct file is "**article_categories.ttl**"<br>They are related using: http://purl.org/dc/terms/subject | |
| Week 7 (June 16 - June 22) | Extend the above code to make a lemon model.<br><br>End of First Phase:<br>● download and extract the necessary files<br>● made a .tsv file<br>● make a lemon file using the code | I have extended the code to make lemon file. The code now makes the lemon file (not from the .tsv file, it makes it independently without using the .tsv file).<br><br>I have tested the lemon file with rapper-utils, there is just one syntax error. | I have tested the lemon file with rapper-utils and here is the error report:<br>rapper -gc abbreviation_lemon_en.ttl rapper: Parsing URI file:///home/div/Desktop/lemon%20file/abbreviation_lemon_en.ttl with parser guess<br>rapper: Guessed parser name 'turtle'<br>rapper: Error - URI file:///home/div/Desktop/lemon%20file/abbreviation_lemon_en.ttl:296 - syntax error at '_'<br>rapper: Parsing returned 867 triples<br><br>Actually the problem is due to links like :"<http://it.dbpedia.org/resource/A.I._-_Intelligenza_artificiale>," which contain '_' |

| | | | |
|---|---|---|---|
| | | | and I can not remove this '_' as then the link will become invalid. When I removed such links there is no error:<br>rapper -gc abbreviation_lemon_en.ttl<br>rapper: Parsing URI file:///home/div/Desktop/lemon%20file/abbreviation_lemon_en.ttl with parser guess<br>rapper: Guessed parser name 'turtle'<br>rapper: Parsing returned 1060 triples |
| Week 8 (June 23 -June29) | Deploy the software on the server, that is:<br><br>Set up a script that:<br>-automatically downloads language specific DBpedia files needed by the extraction for one language<br>-automatically imports these files into virtuoso<br>-runs the extraction on it and exports lemon and tsv files<br>-clears the graph in virtuoso after the extraction has ended<br>-deletes the dbpedia files downloaded in the beginning<br>-loads the next language from a list of languages to convert and start again at 1<br><br>Like this it will automate the extraction. | setting up the server(with ssh keys) and importing the files on the server | |
| | Some changes in the code extractor_lemonMaker.py (by mentor):<br><br>-Use SPARQL **contains** function instead of a **regex** as regex is a much more costly (and slow) operation.<br><br>-There is some hard coded "langMatches("EN")" in queries.<br><br>-remove the dependence | | |

| | | | |
|---|---|---|---|
| | on "only_roman_chars"<br><br>-change the Namespace of lemon to "http://lemon-model.net" instead of monnetproject.eu.<br><br>-change "@prefix : http://www.example.org/lexicon<br>" to "@prefix : http://nlp.dbpedia.org/abbrevbase"<br><br>-Lexicons should have uris of the form http://nlp.dbpedia.org/abbrevbase/lexicon/$lang | | |
| Week 9 (June 30 - July 6) | | I have done the changes in the code as required.<br><br>-corrected langMatches<br><br>-removed the dependence on "only_roman_chars".<br><br>-changed the Namespace of lemon to "http://lemon-model.net" instead of monnetproject.eu.<br><br>-changed "@prefix : http://www.example.org/lexicon<br>" to "@prefix : http://nlp.dbpedia.org/abbrevbase"<br><br>-changes the querying owl:sameAs and rdf:type from regex to contains.<br><br>-changed the locations of input and output files | |

| | | | |
|---|---|---|---|
| | | and tested.<br><br>also<br>fetched the categories successfully. | |
| Week 10<br>(July 7<br>- July 13) | Executed the code for **'en'** in the server.<br>The final files (i.e lemon and tsv) have been created successfully for 'en' and they are present in **/home/akswadmin/abbrev_extracted/en** . The code is working perfectly fine now and is also automated.<br><br>This script generates a report file **server_report** in **/home/akswadmin/abbrev_repo.**<br>This is a report for 'en' :<br><br>**____Server Running Report_____**<br><br>*****************Current Language: en*****************<br>**----------Files for en downloaded successfully----------**<br>**----------Importing files into virtuoso----------**<br>**----------Files imported----------**<br>**----------Running lemon maker----------**<br>**----------Removing graph----------**<br>**----------Done en----------**<br>**==========**<br>**==========**<br>**en took 12408 seconds**<br>**==========**<br>**==========** | Executed the code for **'en'** in the server.<br>which makes lemon and tsv files. | For other languages like 'fr' and 'hi', there is a problem,<br><br>1. there is no file like instance_types_heuristic_fr.ttl.bz2 in fr (and same goes for other languages) and some even don't have instance_types like http://downloads.dbpedia.org/3.9/bat_smg/ .<br><br>2. Some do not have interlanguage_links_chapters.ttl,bz2 like http://downloads.dbpedia.org/3.9/bat_smg/ |
| | Other corrections in the code:<br><br>-Space in @prefix rdf definition<br>-Missing ";" after "a lemon:Lexicon"<br>-  should be changed to "_"<br>-"lemon:entry" triples don't match the entry resources in the file: | | Found that some abbreviations are invalid in lemon<br><br>1. Those starting with digits (though they can contain digits)<br><br>2. Those containing |

| | | | |
|---|---|---|---|
| | $Lexicon lemon:entry :AD_Entry but <A.D._Entry> a lemon:LexicalEntry . should be $Lexicon lemon:entry :A.D._entry :A.D._entry a lemon:LexicalEntry . | | symbols like =, ( ), %, ',¿, ... , & etc<br><br>There is also a problem in the extraction code in regard to URL formatting. There is need to do is append a "<" and a ">" at the beginning and the end when added as objects to triples instead of string replace.It introduced problems in the RDF, because URLs can contain "," and by doing replace(",",">").<br><br><http://dbpedia.org/resource/Category:University_of_California>,_Irvine_alumni><br><br>Instead, it should be the single URLs and append the brackets before adding it to the string. |
| | - Updated the code to replace : lemon:entry :AD_Entry to lemon:entry :A.D._entry , :A.I.entry, ... i.e. in their original form<br><br>-Made a function that will append "<", ">" as required | The strings should not be contained in any URI if format them like :A.D._entry. Besides that, digits, parenthesis and ampersands are allowed in URIs<br><br>For the missing files, accept that there might be data missing. Most important are the abbreviation strings and their full meaning.<br>So included exception | I. On replacing :AD_Entry with :A.D._Entry, the rapper gives a syntax error. else no error.<br><br>II. If there is any symbol in :AD_Entry like :AD%_Entry, :A!D_Entry etc then also rapper gives a syntax error. There are several abbreviations which contain symbols as like **:%3F_Entry , :%5C_Entry , :Negima%3F_Entry** |

| | | | |
|---|---|---|---|
| | | error handling in the code. | III. If there is any entry starting with a digit then also rapper gives a syntax error like :0_Entry<br><br>IV. On changing <A.D._Entry> a lemon:LexicalEntry . to :A.D._entry a lemon:LexicalEntry . then also there is a syntax error. |
| | -Executed the latest code in the server for just 5 abbreviations and the result is as follows:<br><br>root@abbreviationBase:~/en# rapper -gc abbreviation_lemon_en.ttl<br>rapper: Parsing URI file:///home/akswadmin/en/abbreviation_lemon_en.ttl with parser guess<br>rapper: Guessed parser name 'turtle'<br>rapper: Parsing returned 2806 triples | Outdated rapper, so files verified on server and the rapper now gives a correct report which clears the problems I, III and IV.<br><br>Using full URIs<br><http://nlp.dbpedia.org/abbrevbase/lexicon/en><br>a lemon:Lexicon ;<br>lemon:language "en" ;<br>lemon:entry<br><http://nlp.dbpedia.org/abbrevbase/lexicon/en/entry/A.D.> .<br><br><http://nlp.dbpedia.org/abbrevbase/lexicon/en/entry/A.D.><br>lemon:form<br><http://nlp.dbpedia.org/abbrevbase/lexicon/> | Tested on a similar file, added %, parenthesis and ampersand in entry elements (filename: /home/akswadmin/en/abbr_lemon_experiment.ttl) and here is the rapper result:<br>root@abbreviationBase:~/en# rapper -gc abbr_lemon_experiment.ttl<br>rapper: Parsing URI file:///home/akswadmin/en/abbr_lemon_experiment.ttl with parser guess<br>rapper: Error - URI file:///home/akswadmin/en/abbr_lemon_experiment.ttl:12 - syntax error at '%'<br>rapper: Guessed parser name 'turtle'<br>rapper: Failed to parse file abbr_lemon_experiment.ttl guess content<br>rapper: Parsing returned 3 triples |

| | | | |
|---|---|---|---|
| | ran the code for 1000 abbreviations and the lemon file I received was completely correct.<br><br>The code has been executed for **en** and the lemon file is perfect as per the rapper.<br><br>here is the rapper validation:<br><br>**root@abbreviationBase:~# rapper -gc en/abbreviation_lemon_en.ttl**<br>**rapper: Parsing URI file:///home/akswadmin/en/abbreviation_lemon_en.ttl with parser guess**<br>**rapper: Guessed parser name 'turtle'**<br>**rapper: Parsing returned 1663426 triples** | en/entry/A.D.#form> ;<br>lemon:sense <http://nlp.dbpedia.org/abbrevbase/lexicon/en/entry/A.D.#sense> ;<br>a lemon:LexicalEntry .<br><br><http://nlp.dbpedia.org/abbrevbase/lexicon/en/entry/A.D.#form><br>lemon:writtenRep "A.D."@en ;<br>a lemon:LexicalForm .<br><br><http://nlp.dbpedia.org/abbrevbase/lexicon/en/entry/A.D.#sense><br>lemon:definition [<br><br>lemon:value "Anno Domini"@en<br>] ;<br>lemon:reference <http://dbpedia.org/resource/Anno_Domini> ;<br>a lemon:LexicalSense .<br><br>This solves the issue as chars and digits are allowed in URIs. | (also checked individually for %, ( ) , & and , and all of them resulted into syntax error.)<br><br>-An error due to **lemon:value "Oran " Juice" Jones"@en** .<br>Its due to multiple " ", so corrected it using **"Oran \"Juice\" Jones"@en**<br><br>-On executing the code for all the abbreviations then there might be some error due to which the code stops<br><br>-Problem solved using exception handling:<br><br>**Index Error** when converting type, category etc to string. The error was obtained when any of them were empty i.e. when the query did not return the result for them, so I used exception handling to remove it.<br><br>Then there were errors due to not initializing variables rdf_type, |

| | | | |
|---|---|---|---|
| | | | label, category, owl_sameAs, ref which gave an error when again the query was not able to return the results to the respective query. |
| **Second Phase Completed** | **As a next task, run the script for**<br><br>**de, nl, fr, es, it, ru, cs, pl, sv, da, et, pt, no, nn, sl, fi** | | |
| Week 11 (July 14 - July 20) | | completed extraction for 'en', the lemon file made is absolutely correct as per rapper validation.<br><br>Executed the code for other languages in the database. | getting significant differences in the count of abbreviations between my extraction and the extraction mentor originally did.<br><br>For German, tsv has 4689 lines, mine only has 696.<br><br>For French and Dutch, you extract around 300 lines less. |
| | Get the URIs right<br><br>Abbreviations containing "_" should not be included at all<br><br>In the Polish and French abbreviations, there are URL encoding (%3F instead of ?). If encounter "%" in the abbreviations, run a URLdecode function over it. | The problem of incomplete abbreviations was that for those abbreviations whose query returned nothing, the dictionary **abbrevs** was no formed, thus it was not written in the file. I just included a condition which prevents the above problem.<br><br>**http://localhost:8890/sparql** | there are abbreviations in there that contain "_", thus massively increasing the scope. For example, in the German file, there are a lot of dates. Also something does not work right with the URIs:<br><br>2980er_v._Chr. 30. Jahrhundert v. Chr. "30. Jahrhundert v. Chr."@de <ttp://dbpedia.org/resource/30._Jahrhundert_v._Chr> |

| | | | |
|---|---|---|---|
| | | query this endpoint for every language, to remove problem of missing data. | missing "h" at the start of the URI and the missing period at the end of the URI. From what I have seen, it only happens to URIs that contain "_".<br><br>missing some disambiguations for the abbreviation "A.A." in the German file. Disambiguations. |
| | I produced an overall 22859 abbreviations with 78197 meanings in 99 languages. Overall unique abbreviations (aggregated over the languages) are 15646.<br><br>The other languages (20 out of 119) for which abbreviations could not be extracted had either inappropriate data, or data not desired.<br><br>deleted the old extractions, the dbpedia files etc<br><br>clear the virtuoso graph, make a new graph "http://nlp.dbpedia.org/abbrevbase/" and import all abbreviations from all languages into it. | Removed all the abbreviations which only have digits and no character. | we have downloaded / instance_types_heuristic_$language.ttl.bz2 for instance types. But some languages do not have instance_types_heuristic_$language.ttl.bz2 rather they have instance_types_$language.ttl.bz2 so we need to download both the files (which ever available) for all the languages. |
| **Third Phase Completed** | **I produced an overall 22859 abbreviations with 78197 meanings in 99 languages. Overall unique abbreviations (aggregated over the languages) are 15646.** | | |