Hunting for Direct Translations across Wikipedia Articles

Duc Manh Hoang and Marco Ronchetti
Department of Information Engineering and Computer Science
Università di Trento
Povo di Trento, Italy
email:ducmanh.hoang@studenti.unitn.it, marco.ronchetti@unitn.it

***Abstract*—This paper deals with a problem in the area of Cross-Language Plagiarism Detection. In particular, it presents a system, able to detect portions of a Wikipedia page, which have been obtained by translating a Wikipedia page on the same semantic content but written in a different language. The problem is relevant in the context of Wikipedia pages maintenance, and could be of interest in other areas such as news comparison in different languages. We discuss the problem, the system and its implementation and briefly present its evaluation.**

Keywords-Machine translation; Wikipedia; Plagiarism.

I. INTRODUCTION

Although the term "Encyclopedia" was first used in the XVI-th century, the most important attempt to code all the mankind knowledge under such name happened in the mid of the XVIII century. It was coordinated by Denis Diderot and had a very important influence on the development of the Age of Enlightenment, which shaped the western culture for the years to come [1]. In our lives, we witnessed another cultural revolution connected with the Encyclopedia concept: the birth and growth of Wikipedia, the largest cultural collaborative writing effort in mankind history. The importance of Wikipedia cannot be overstated. According to Alexa [2], it ranks 5th among the most visited Web sites, and is the first non-commercial one, being surpassed only by Google (www.google.com), YouTube (www.google.com), Facebook (www.facebook.com), and Baidu (www.baidu.com).

Semantic information can be extracted from Wikipedia: the DBPedia initiative pioneered such effort [3], allowing to build semantic applications on top of it (see, e.g., [4-6]).

But Wikipedia is not just "one" collaborative Encyclopedia. It is rather a collection of many versions in different languages: presently 295 (but 10 do not reach 100 pages). The English Wikipedia contains more than 5 million articles, while versions in 12 other languages exceed 1 million articles, and 124 more languages contain at least 10.000 entries [7]. Recognizing the importance of the multilingualism, Wikipedia offers special links among pages dealing with the same topic, but written in different languages: the so-called "interlinks". Interlinks allow users to easily browse the corresponding pages in other languages, and hence to compare and integrate the knowledge contained in a page with the one of the other Wikipedia versions (provided the different language is not a hurdle for the curious reader). In fact, it is quite natural that some entries are richer in a language that in another, as this reflects a "national interest". For instance, a type of German locomotive not having a special historical value is (probably well) documented in the German Wikipedia, but hardly mentioned in other languages or, when mentioned, the article in languages other than German will probably be sketchy and contain only some of the most important details. In spite of this, non-German railway historians will nonetheless be interested in finding out more.

A comparison of pages in different languages is also useful for editors, who wish to integrate a page in a language, gathering knowledge via the interlink.

In an attempt to increase the number of pages (especially for languages with a limited coverage), Wikipedia has been recently promoting the translation of pages, which exist in the "main editions" and are absent in other languages. Since there exists a procedure for acknowledging that a page has been translated [8], such form of "plagiarism" has no negative connotation.

It is interesting for various reasons to find out if a Wikipedia page in a given language has been (partially or totally) translated in other languages. We therefore asked ourselves, if there is a way to automatically detect such translations. For instance, it could help identifying semantic difference between papers in different languages (e.g, missing parts) and could be used to automatically signal the necessity or opportunity to improve a page in a given language.

We developed a software tool to deal with such problem. In the present paper, we describe its architecture and working mechanism, and present a sample of the results obtainable with it. Section II presents the relation of our work with the area of Cross-Lingual Plagiarism Detection; Section III discusses how we decided to attack the problem of comparing Wikipedia pages on a given topic, written in different languages; Section IV describes the process of comparing the pages; Section V presents the overall software architecture. In Section VI we briefly presents the evaluation, and finally in Section VII we draw our conclusions.

II. RELATION WITH CROSS-LINGUAL PLAGIARISM DETECTION

Our problem has some common traits with Cross-Lingual Plagiarism Detection (CLPD), which has been studied by several authors (see, e.g., [9-12]). There are, however,

 ISBN: 978-1-61208-600-2

differences. Plagiarism Detection aims at finding whether a "suspect" document contains parts of text taken by any document on the Web. In the multilingual case, the problem is exacerbated by the difficulty of finding a set of candidate sources written in other languages, so that the simple strategy of using traditional search engines is not effective. Then, also the comparison of suspect and potential source is more difficult since it has to be performed across languages.

A typical architecture for CLPD [10] comprises heuristic retrieval (i.e., the gathering of possible sources), detailed analysis (to compare the suspect with every document collected by the retrieval) and heuristic post-processing (for merging or discarding possible sources).

Our case is simpler, since our set is predefined by semantics, and it is the set of documents related by interlinks. We can, hence, focus on the second part of the problem, avoiding heuristic retrieval, and have fewer difficulties in dealing with it.

Also, there is another important difference: plagiarism is usually considered as an unacceptable practice. Plagiarists hence often try to disguise the copied parts, e.g., by paraphrasing portions of the text, so as not to be detected by search engines. Instead, in the case we are interested in, copying is a socially accepted and even encouraged practice, which helps spreading the knowledge to other communities, and therefore authors do not need to try to hide it.

Typical strategies for Cross Language analysis include lexicon-based systems, thesaurus-based systems, comparable corpus-based systems, parallel corpus-based systems and machine translation-based systems. We cannot discuss all of them here, as the area is wide, and refer the reader to [13]. The approach we chose, which is machine translation-based, is described in the following sections.

III. CONSIDERING WIKIPEDIA PAGES WRITTEN IN DIFFERENT LANGUAGES: THE NORMALIZATION PROCESS

The problem to solve is to be able to compare a pair of corresponding pages in different versions (i.e., languages) of Wikipedia: say P^x_Y and P^x_z (Page X in Language Y and Page X in Language Z). The way to compare a pair of pages could be to try to extract the contained semantic information, mapping it to an ontology and comparing it. We think that such an approach is bound to fail. In fact, since both P^x_Y and P^x_z are about x, the semantic meaning obviously matches. The richness of the semantics could be different (as one could contain more details than the other), but even if the richness is the same, this does not imply that a page is the translation of the other. More information about the structure and actual content of the page has to be taken into account.

Such information must come from the texts we want to compare, but they are in different languages. To make them comparable, we decided to translate them. In order to make our approach scalable, we opted to use automatic translation. We were well aware of the limits that today's machine translation (MT) has, but decided anyway to give it a try to verify if, in spite of them, the approach could work.

Having to compare a German and a French page on the same topic (P^x_G and P^x_F) we could decide to translate one of them in the other language, and then compare say $P^x_{G->F}$ and P^x_F, where the suffix X->Y means page "written in language X and translated into language Y". This introduces an asymmetry, so we could also compare P^x_G and $P^x_{F->G}$ and then match the two results.

However, we thought that such an approach would have presented some problems. First, we were interested in checking not only two languages, but a set of the largest Wikipedia versions (namely, we chose English, French German and Italian). This would have implied multiple translations. Second, the quality of publicly and freely available MT engines seems far from being uniform when translating between languages. Since the technology used by engines, such as Google Translate is considered a trade secret, it is difficult to find evidence in academic papers on what is going on behind the scenes. There are of course reviews of MT techniques (such as, e.g., [14]), and indications that Google uses "mostly" statistical methods [15], which make unnecessary to "bridge" though an intermediate language or model. In any case, the quality of translation into English seems to be better than the one into other target languages, maybe also because its grammar is far simpler that the one of many other languages, including the ones we have chosen for our exercise, or because of a larger base, since English is today's *lingua franca*. We cannot prove this assumption, as we did not find scientific evidence for this fact. However, combining the combinatorial problem with the guess that translation into English is at least not worse than translations into other languages, we decided to "normalize" all the texts (written in other than English languages) by translating them into English. Hence, for every topic X we are interested in, we consider the set $\{P^x_E, P^x_{F->E}, P^x_{G->E}, P^x_{I->E}\}$.

We therefore wrote a software component which, given a Wikipedia page in one of the four languages, checks if the interlinks into the other three languages are present, and once they are found it performs the needed translation. We could use several MT engines (Bing, Google, SDL, Yandex). According to evaluations available on the Web, they seem to provide similar performances. Once again, we were facing the impossibility to base our work on scientifically sound grounds, but had to trust information which, in spite of being rather coherent, does not offer scientific rigor. In the end we decided to use the Yandex API [16] to perform the translation, since they were the most inexpensive available option (with up to 2 million characters/month free, and the cheapest option above that threshold).

IV. COMPARING THE PAGES

For a given topic X, we now have four documents: P^x_E, $P^x_{F->E}$, $P^x_{G->E}$, $P^x_{I->E}$. To compare the pages, we first segment the text by breaking their content into sentences. We use a list of abbreviation to avoid getting confused by the punctuation used for abbreviations rather than for ending sentences.

The next step is to apply to each sentence N-Gram segmentation, a technique for breaking a stream of text into units of N ordered adjacent words [17]. Part–of-speech (P.O.S.) tagging is then applied to identify the role of each word in the sentence (e.g., noun, verb, adjective etc.). P.O.S.

 ISBN: 978-1-61208-600-2

tagging is needed to perform the next operation, which is lemmatization: a technique similar to stemming but aware of the context in which a word is situated. This allows replacing, e.g., "better" with "good", verbs (such as I "am") into their infinitive form ("be"), etc. Stop words (such as articles, but also any very frequent word) are then removed: since they are very common, their presence in unrelated sentences would generate noise in terms of false positives when comparing their content, so it is better not to have them in the text (even if by doing so some relevant part of "meaning" gets omitted).

At this point, each of the four normalized documents have been exploded in a set of cleaned-up sets of words $\{S^x_{Li}\}$, where L stands for the original language (although all documents now contain only English words) and i is the index of the phrase in the document. The documents P^x_L, which are at the origin of our sets, generally have different number of sentences, which we will call N^x_L, so for each S^x_{Li} the index i runs from 1 to N^x_L.

Let us now try to ascertain that a portion of document P^x_A has been copy-translated into P^x_B, or vice versa. We can examine pair or sentences, but we cannot make assumptions on where they are: a portion from the beginning of a document could have been copied onto the central part of the other, so we need to compare each sentence in P^x_A with every other sentence in P^x_B. This will generate a matrix of dimension N^x_A x N^x_B, in which the cell (i,j) contains a number representing a measure of similarity between the sentences S^x_{Ai} and S^x_{Bj}.

We now need to know how such measure is computed, and what can we do with the matrix.

To evaluate sentence similarity, we tested two different approaches: we used both Cosine similarity [18] and Jaccard similarity [19]. For each pair of sentences $\{S^x_{Ai}, S^x_{Bj}\}$ we build a bag of words, containing all the words which appear in at least one of the two sentences (but each word is present only once in the bag, regardless of the actual number of occurrences in the sentences). The words are ordered (in an arbitrary way), defining in this way an M-dimensional space, where M is the cardinality of the bag of words. For each sentence, we can then compute its position in such vector space: the number of occurrences of the z-th word in it gives the value of the z coordinate. Having the coordinates of the two sentences, their Cosine similarity is evaluated as the scalar product between the vectors, which represent them (such value is in the interval [0,1], since only the positive subspace is considered, as the number of occurrences which determine the coordinates cannot be negative).

The Jaccard Similarity is instead computed as the ratio between the cardinality of two sets: $|S^x_{Ai} \cap S^x_{Bj}| / |S^x_{Ai} \cup S^x_{Bj}|$. This value is also in the interval [0,1].

At this point we forked our project, using these two different measures of similarity (Cosine and Jaccard) and then proceeding in the same way. In both cases, we end up with a score matrix for topic X and the pair of languages {A,B}, and in both cases the values of the cells in the matrix are numbers between 0 and 1.

The closest a cell is to one, the highest the similarity between the two corresponding phrases. However, in the Wikipedia page generation case, a "copy-translate" is not just related to one single sentence, but rather to a section of the paper, which consists of multiple adjacent sentences, each with a high similarity value. Hence we are interested in detecting diagonal subsets with high similarity values in the score matrix. For instance, we are interested in finding situations where not only S^x_{Ai} and S^x_{Bj} are similar, but also the pairs $\{S^x_{Ai+1}, S^x_{Bj+1}\}$, $\{S^x_{Ai+2}, S^x_{Bj+2}\}$, …, $\{S^x_{Ai+n}, S^x_{Bj+n}\}$.

To facilitate the identification of such sequences, we canceled the noise, by putting to 0 all the cells, which have a value less than a given threshold. To define the threshold level, we considered how the data are distributed in the matrix, and assumed a Gaussian distribution for the noise. We kept only the tail of the high values. We then looked for diagonals sequences: these reveal portions of the text, which are very similar and hence are likely to be copy-translated.

V. OVERALL ARCHITECTURE

We summarize the architecture of our system, which is outlined in Figure 1. A harvester (Document Reader) gets the documents and generates a set composed by a given Wikipedia page in one of the four languages and the interlinked pages in the other three languages.

It then translates all the non-English pages into English, obtaining a set of quadruples $\{P^x_E, P^x_{F->E}, P^x_{G->E}, P^x_{I->E}\}$. More details about the Document Reader are given later.

Given the quadruple, each of its documents is passed to the Preprocessing Unit, which performs text segmentation (into phrases), P.O.S. tagging, lemmatization and stop words removal.

The result is given to the Text-Similarity Unit, which evaluates Cosine similarity and Jaccard similarity between each pair of sentences contained in each pair of elements of the quadruple.

The output of the Text-Similarity Unit is passed to the Post Processing Module. Here, for each pair (P^x_A, P^x_B) where A and B are two different elements of the set {E, F->E, G->E, I->E), the values computed by the text-similarity unit compose the score matrix, which is cleaned discarding the low values, and for which non-null diagonal sequences are searched. If some diagonals are found, we have a candidate "copied" section of the articles. Of course, similarity is symmetric, so we still need to know which is the original, and which the copy. This can be easily understood by checking the version in Wikipedia history. The task is hence accomplished, and we can pass to the evaluation phase.

 ISBN: 978-1-61208-600-2

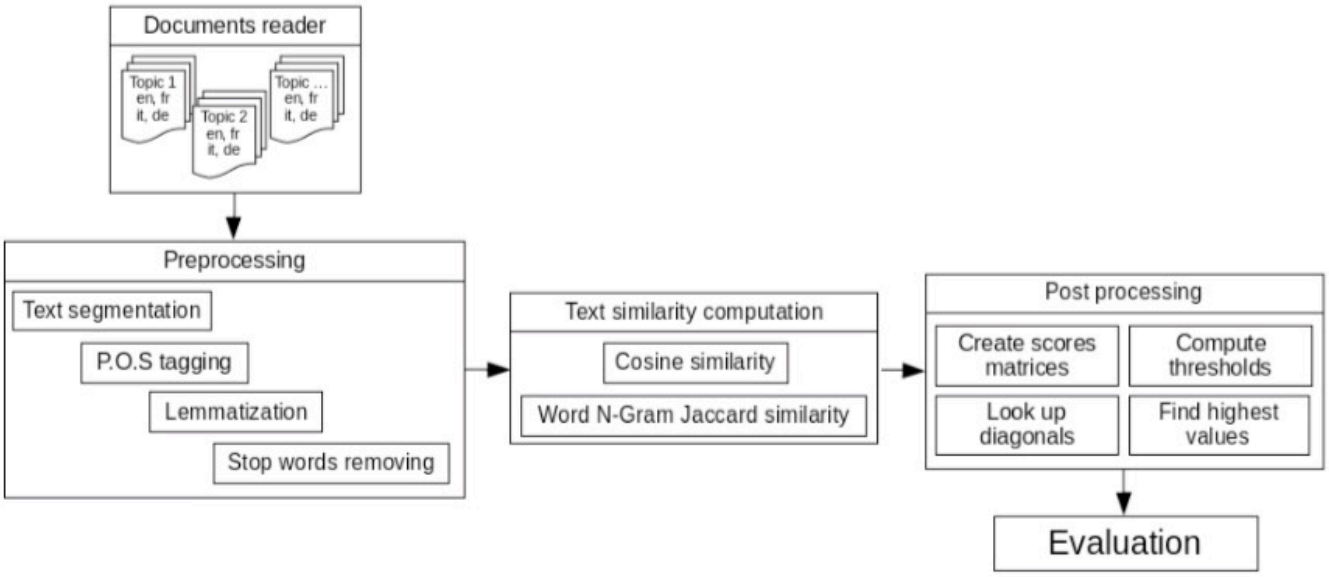

Figure 1. Overall logical architecture of the system (see text for a description).

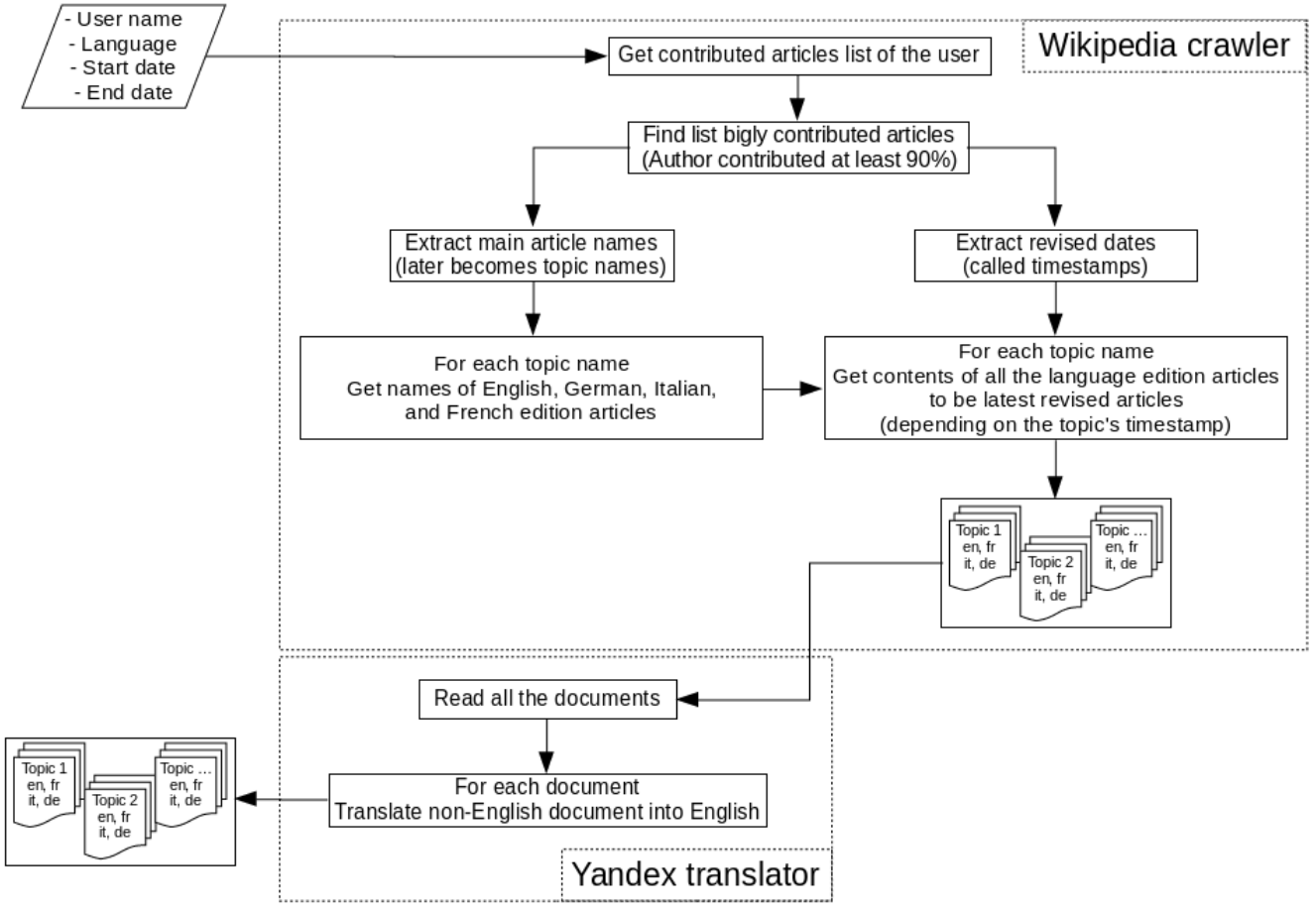

Figure 2. Explosion of the Documents Reader component (see text for a description).

Let us now come to a more detailed description of the harvester (Document Reader), which is exploded in Figure 2. It is composed by the Wikipedia Crawler and a Translation Unit. The Wikipedia Crawler is fed with the specification of the user name of a Wikipedia author, a language and a time span running from a start date to an end date. The Crawler extracts the list of all the Wikipedia articles (in the given language) that the user contributed to in the selected time span, and selects those for which the author was the major contributor (according to a customizable percentage parameter). For these, the interlinked articles are retrieved. Article revisions are considered to make sure that the compared versions refer to the same time. This is very important, since an article, which was used as a source for a copy-translation, could have been modified after the translation was performed.

Once the set of four documents has been generated, it is necessary to translate the Italian, French and German ones into English. In our implementation this is done with the Yandex translator, but the Translation Unit could use any other translator.

The main software tools we use are the Wikipedia API [20], DKPro Core Library [21], the UIMA-Unstructured Information Management Architecture [22] and the already mentioned Yandex API.

VI. EVALUATION

Evaluating the results has been a costly operation, since we need an "oracle" able to give us a human evaluation of whether a portion of a page has been copy-translated. The time needed to perform such an operation is non-negligible, and the results are not always clear-cut. Sometimes a portion of the document is not simply translated, but reworked and paraphrased. In other cases, semantic identity of the content pushes the authors to write very similar sentences, even when being unaware of each-other's work: one has to remember that the topic of the considered pages in a given set is the same, and hence what is classified ad "paraphrase" could well simply be due to "semantic similarity".

We used 56 topics containing 148 pages generated by 4 authors. Author 1 translated some pages from Italian Wikipedia to the French one. Author 2 translated from German Wikipedia to the Italian one. Author 3 translated from English to French. Author 4 did not rely his/her contributions on copy-translation.

Our (human) judgment of these authors is reported in Table 1.

TABLE I. EVALUATION SET

Author	Total pages	Partially copied pages	Paraphrased pages
Author 1	33	7	10
Author 2	30	10	4
Author 3	59	21	5
Author 4	26	0	5

For each pair P^x_A, P^x_B, we manually evaluated whether the version of the pages in other languages had significant, similar sections (we will call this "oracle evaluation"). We categorized the pairs of pages (a topic in two languages) by using three descriptors: copied, paraphrased, not copied (where a page is considered to be "copied" if a significant section of it (at least 4 or 5 sentences) is similar to a page written in a different language.

We then compared the human annotated results with the predictions of our system, and checked if there was a full agreement (both systems stating the same thing), partial agreement (the machine declaring that there was a copy, and the human describing the mapping as a paraphrase) or no agreement (oracle and machine producing opposite statements). The possible cases and the corresponding results are reported in Table 2.

TABLE II. EVALUATION RESULTS

Oracle	Prediction	Evaluation	Numerosity
YES	YES	True positive	31
YES	NO	False negative	10
NO	YES	False positive	1
NO	NO	True negative	82
Paraphrase	YES	Uncertain	7
Paraphrase	NO	Uncertain	17

In 16% of the cases we examined, the oracle was uncertain whether there had been a copy-translation between the considered pair of documents.

Out of the cases where the oracle decided with certainty for the NO, the system prediction was right 99% of times .
Out of those where the oracle decide with certainty for the YES, the system prediction was right 75% of times.

Seen from a different perspective and taking into account also the cases when the oracle was uncertain, whenever the system predicted the presence of copy-translation, it was right 79% of times. When it predicted its absence, it was right 75% of times.

We find no difference in using the matrices obtained using Cosine similarity and Jaccard similarity: both measures yield results of the same quality.

VII. DISCUSSION AND CONCLUSIONS

We presented our work on finding whether a Wikipedia page originated from another one, written on the same topic but in a different language, by translating a portion of the page. The work is somehow close to the domain of Cross-Language Plagiarism Detection, but presents some peculiarity, which distinguishes it from the mainstream in that area.

The work can be the basis for tools, which could be useful for Wikipedia maintainers, and could be used for statistical analysis of the Wikipedia body of knowledge. For instance this work, given a Wikipedia author, could help classifying her/his type of contributions.

The evaluation of the system we developed shows a very good reliability in a domain, where even humans have difficulty to establish with certainty the truth.

 ISBN: 978-1-61208-600-2

In future, it would be interesting to examine if our approach also works with other languages, such as the Asian ones.

The developed software has been released in public domain and is publicly available at [23]. Some more detailed explanations are available there in the readme file, which also reports a sample of the experiment. For any additional clarification, interested people are invited to contact the authors.

REFERENCES

[1] P. Blom, "Enlightening the world: Encyclopédie, the book that changed the course of history", New York: Palgrave Macmillan (2005)

[2] https://www.alexa.com/topsites, last visited Sept. 6, 2017

[3] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "DBpedia: A nucleus for a Web of open data." The semantic Web (2007): pp. 722-735.

[4] A. Prato, and M. Ronchetti, "Using Wikipedia as a reference for extracting semantic information from a text." Advances in Semantic Processing, 2009. SEMAPRO'09. Third International Conference on. IEEE, 2009. pp. 56-61

[5] F. Valsecchi and M. Ronchetti, "Spacetime: a two dimensions search and visualisation engine based on linked data." Conference on Advances in Semantic Processing (SEMAPRO). 2014.

[6] M. Battan and M. Ronchetti, "QwwwQ: Querying Wikipedia Without Writing Queries." International Conference on Web Engineering. Springer, Cham, 2016. pp. 389-396

[7] https://www.wikipedia.org/ last visited Sept. 6, 2017

[8] https://en.wikipedia.org/wiki/Wikipedia:Translation, last visited Sept. 6, 2017

[9] C. K. Kent and N. Salim, "Web based cross language plagiarism detection." Computational Intelligence, Modelling and Simulation (CIMSiM), 2010 Second International Conference on. IEEE, 2010. pp. 199-204

[10] A. Barrón-Cedeño, P. Gupta, and P. Rosso, "Methods for cross-language plagiarism detection." Knowledge-Based Systems 50 pp. 211-217, 2013.

[11] M. Franco-Salvador, P. Gupta, and P. Rosso, "Cross-language plagiarism detection using a multilingual semantic network." European Conference on Information Retrieval. Springer, Berlin, Heidelberg, 2013. pp. 710-713

[12] E. Nava, F. Wm. Tompa, and A. Shakery, "Using a dictionary and n-gram alignment to improve fine-grained cross-language plagiarism detection." Proceedings of the 2016 ACM Symposium on Document Engineering. ACM, 2016. pp. 59-68

[13] M. Potthast, A., Barrón Cedeño, B., Stein, and P. Rosso, "Cross-language plagiarism detection. Language Resources and Evaluation (LRE)", Special Issue on Plagiarism and Authorship Analysis 45 (1), pp. 1–18. 2011.

[14] S. Tripathi and J. K. Sarkhel, "Approaches to Machine Translation", Annals of Library and Information Studies, Vol 57, pp. 388-393, 2010.

[15] T. Mikolov, V. Le Quoc, and I. Sutskever, "Exploiting similarities among languages for machine translation." arXiv preprint arXiv:1309.4168 (2013).

[16] https://tech.yandex.com/translate/, last visited Sept. 6, 2017

[17] P. Norvig, "Natural language corpus data". In Beautiful Data, edited by T.Segaranand and J.Hammerbacher, pp. 219–242. Sebastopol, Calif.: O'Reilly (2009)

[18] A. Singhal, "Modern Information Retrieval: A Brief Overview". Bulletin of the IEEE Computer Society Technical Committee on Data Engineering 24 (4): 35–43 (2001)

[19] P. Jaccard, "The distribution of the flora in the alpine zone", New Phytologist, 11: 37–50 (1912)

[20] https://www.mediawiki.org/wiki/API:Main_page, last visited Sept. 6, 2017

[21] https://dkpro.github.io/dkpro-core/, last visited Sept. 6, 2017

[22] https://uima.apache.org/, last visited Sept. 6, 2017

[23] https://github.com/ducmanhhoang/Wikipedia-Matching, last visited Sept. 6, 2017

 ISBN: 978-1-61208-600-2