HAL
open science

A Multilingual Approach to Discover Cross-Language Links in Wikipedia

Nacéra Bennacer Seghouani, Mia Johnson Vioulès, Ariel López Maximiliano, Gianluca Quercini

To cite this version:

Nacéra Bennacer Seghouani, Mia Johnson Vioulès, Ariel López Maximiliano, Gianluca Quercini. A Multilingual Approach to Discover Cross-Language Links in Wikipedia. 16th International Conference Web Information Systems Engineering (WISE), Nov 2015, Miami, United States. 10.1007/978-3-319-26190-4_36 . hal-01276205

HAL Id: hal-01276205

https://hal.science/hal-01276205v1

Submitted on 11 Apr 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Multilingual Approach to Discover Cross-language Links in Wikipedia

Nacéra Bennacer, Mia Johnson Vioulès,
Maximiliano Ariel López, and Gianluca Quercini

CentraleSupélec - LRI, 91192 Gif-sur-Yvette, France
nacera.bennacer@lri,mia.vioules@student.ecp.fr,
maximiliano.lopez@student.ecp.fr,gianluca.quercini@lri.fr

Abstract. Wikipedia is a well-known public and collaborative encyclopaedia consisting of millions of articles. Initially in English, the popular website has grown to include versions in over 288 languages. These versions and their articles are interconnected via cross-language links, which not only facilitate navigation and understanding of concepts in multiple languages, but have been used in natural language processing applications, developments in linked open data, and expansion of minor Wikipedia language versions. These applications are the motivation for an automatic, robust, and accurate technique to identify cross-language links. In this paper, we present a multilingual approach called *EurekaCL* to automatically identify missing cross-language links in Wikipedia. More precisely, given a Wikipedia article (the source) *EurekaCL* uses the multilingual and semantic features of BabelNet 2.0 in order to efficiently identify a set of candidate articles in a target language that are likely to cover the same topic as the source. The Wikipedia graph structure is then exploited both to prune and to rank the candidates. Our evaluation carried out on 42,000 pairs of articles in eight language versions of Wikipedia shows that our candidate selection and pruning procedures allow an effective selection of candidates which significantly helps the determination of the correct article in the target language version.

1 Introduction

Over the last 14 years, Wikipedia has grown to become the largest online encyclopaedia to date and one of the most popular websites worldwide. Its more than four million articles in English alone describe a wide range of topics, but the most interesting feature of this collaborative effort is that its vast amount of information is linked across more than 200 languages. For example, the article titled "Decision Theory" in the English Wikipedia has a cross-language link to the pages "Teoría de la decisión", "Teoria della decisione" and "决策论" in the Spanish, Italian, and Chinese versions of Wikipedia respectively. Over the past few years, attention has turned to this network and efforts have been made to build more robust and accurate links between Wikipedia versions. At its origin, cross-language links in Wikipedia were introduced to help users further their understanding by exploring concepts in multiple languages. However, as Wikipedia

continues to grow, these links have been the means to achieve many other goals, including enriching linked open data platforms [2], introducing automatically new intra-language links in Wikipedia articles [10], and promoting cross-language information retrieval applications [1, 3, 9]. Typically, the cross-language links are manually added by authors of articles and are subject to being incomplete or erroneous due to the lack of an automatic verification tool. When the author of an article does not link to another Wikipedia language version for a specific reason, this is called a *missing cross-language link*. For instance, as of May 2015, the article titled "Caffettiera" in the Italian Wikipedia has a missing cross-link to the corresponding articles in French ("Cafetière") and Spanish ("Cafetera").

The goal of our paper is to find such articles with missing cross-language links and determine their appropriate respective articles in other languages, a challenging task which has already been addressed by other researchers [5, 7, 8]. Compared to existing approaches, we noticed a serious limitation in each algorithm at the stage of selecting the set of candidate articles in the desired target language (candidate set). Moreover, all these algorithms are language dependent. For instance, the algorithm defined in [8] needs to train a different SVM for each pair of languages under consideration. In [7] language features are used to determine the named entities in Wikipedia. In [5] language dependent textual features are also used to determine the number of common words belonging to two different languages.

In this paper, we present a multilingual approach called *Eureka Cross-Language Link* (*EurekaCL*) to automatically identify missing cross-language links in Wikipedia. *EurekaCL* works with a given article in a source language to identify the best possible corresponding articles in a target language. Moreover, *EurekaCL* is language independent in the sense that the algorithm needs no change or parameter tweak to be applied to any pair of language versions of Wikipedia. These languages can also be heterogeneous and based on different alphabets. For example we can search cross-links between Chinese and Greek articles.

Given a Wikipedia article (the source) *EurekaCL* uses the multilingual and semantic features of BabelNet 2.0 [6] in order to effectively identify a set of candidate articles in a target language that are likely to cover the same topic as the source. The resulting candidate set is further enhanced by exploiting the Wikipedia categories and the cross-language link paths between articles and is then reduced by an effective pruning procedure, which eliminates those candidates that are already connected via a path of cross-language links to other articles in the source Wikipedia. As a result, the size of the resulting candidate set is reasonably small while containing almost always the correct target article. Each candidate is assigned a score which reflects either the amount of paths of cross-links that connect it to the source or the size of their common neighbours. The score is then used to produce a ranked list of candidates.

Unlike existing approaches, which are usually tested on around 1,000 source articles, we evaluate *EurekaCL* on a dataset of 42,000 source articles. Moreover, the multilingual nature of *EurekaCL* allows us to evaluate it on a dataset including articles on multiple source languages and multiple target languages, while

the other approaches only consider one source language and one target language at the same time. The results that we obtain show that *EurekaCL* performs very well.

The remainder of the paper is organized as follows: Section 2 reviews some previous approaches to discovering missing cross-language links. In Section 3 we outline the preliminary notations used to describe our algorithm, which is then detailed in Section 4. In Section 5, we evaluate *EurekaCL* and we conclude the discussion in Section 6.

2 Related Work

The challenge of finding missing cross-language links in Wikipedia has been addressed from many different angles since the first proposed solution by [8]. Among these different approaches, two main choices resonate: supervised versus unsupervised learning algorithms and graph-based versus text-based approaches. Overall, the comparison between the algorithms is difficult as each set of authors evaluated their algorithms with a different dataset, varying language versions of Wikipedia, and differing definitions of recall.

In [8], the authors opt for a classification-based approach with both graph-based and text-based features in order to find missing cross-language links between the German and English versions of Wikipedia. Their features rely heavily on the *chain link hypothesis*, , which assumes that two equivalent (cross-language linked) Wikipedia articles are connected or should be connected via a chain of intra-language and inter-language links. The authors recognize the need to narrow down the candidate set for a source article and keep the top 1,000 candidate articles with the highest number of chain links to the source article. Then, they train their classifier using five graph-based features and two text-based features to predict whether or not a source article has a cross-language link to a candidate. They evaluate their classifier on a dataset of 1,000 source articles (RAND1000) and achieve a precision of 93.5% and a recall of 69.9%. As this approach relies on language features, it requires to train a classifier for each pair of languages. Furthermore, performing the morphological analysis on each article to generate the text-based features is very costly.

Another approach, *WikiCL*, is an unsupervised and graph-based algorithm [7]. Once again, *WikiCL* first reduces the number of possible candidates in the target language. To do so, they take a categorical approach to classify each article as representing: (i) a non-geographic named entity, (ii) a geographic named entity, or (iii) a non-named entity. The candidates for a source article in the target language should lie in the same category as the source article. The candidates are selected using graph-based features then ranked using the *semantic relatedness* which takes into account all links. The evaluation of *WikiCL* uses English as the source language and find corresponding articles in Italian, French, and German (as target languages). The precision ranges from 89% to 94% and recall is between 89% and 93%.

CLLFinder is a supervised classification-based algorithm that uses a mix of four graph-based and text-based features [5]. The *CLLFinder* algorithm also uses the *chain link hypothesis* and adds further candidates in the target language that belong to the respective categories as the source article. Their experiments show that the size of the candidate set is too high. As in [8], the authors sort the articles by the number of times they appear in the set and reduce them to the 1,000 most frequent candidates. These candidates are then submitted to a classifier using graph-based and text-based features including cross-language links, title similarity (with translation) and text overlap (without translation). The cross-language link-based feature consists in looking for an intermediate language version article related via a cross-link both to the source and a candidate article. Their evaluation made on 1,000 articles in Portuguese for which their English counterparts are known (with French, Italian, and Spanish as intermediate languages) shows a precision between 85% and 100% and a recall between 97% and 98%. *CLLFinder* outperformed in precision and recall thanks to the cross-language link transitivity hypothesis.

As in the aforementioned approaches, our purpose is to determine, for a given article in a source language, the best possible corresponding article in a target language. However, *EurekaCL* does not require a training set and is both multilingual and language independent thanks to the use of the lexicographic and semantic knowledge of BabelNet. Moreover, unlike *CLLFinder*, *EurekaCL* completely exploits the transitivity nature of cross-link paths to enhance and prune the candidates and to determine the correct one.

3 Preliminaries

A language version α of Wikipedia is considered a directed graph W_α, where each node n_α represents a Wikipedia concept (article) in the language version α and has a set of categories $Cat(n_\alpha)$. An article n_α (e.g., the one titled "Paris" in the English Wikipedia) is usually connected to other related articles in the same language version (e.g., "Eiffel Tower", "Louvre") via *intra-language links*; an intra-language link between two articles n_α and m_α is denoted with $rl(n_\alpha, m_\alpha)$. Also, an article n_α (e.g., "Paris" in the English Wikipedia) can be connected to articles in other language versions covering the same topic (e.g., "Paris" and "Parigi" in the French and Italian Wikipedia respectively) via *cross-language links*; a cross-language link between the articles n_α and n_β is denoted with $cl(n_\alpha, n_\beta)$. There could be more than one path connecting two articles in different language versions. We denote with $path(n_\alpha, n_\beta)$ the set of paths connecting n_α to n_β considering cross-language links.

Theoretically, the *cl* links should be symmetric and transitive. Moreover, there should be at most one Wikipedia article in a given language version β that is directly linked to a given Wikipedia article n_α via a cross-language link. However, the reality is more complex than that, the cross-links between Wikipedia articles in different language versions could be inconsistent.

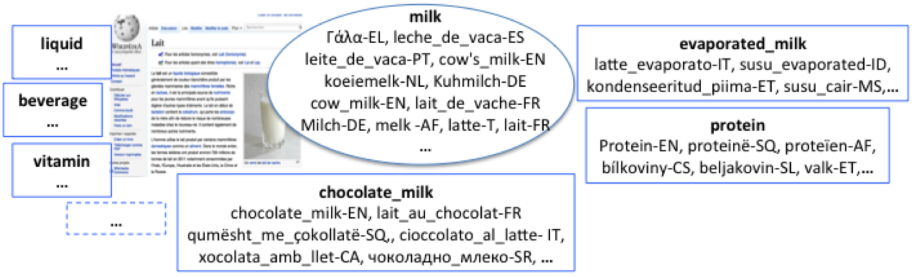

Fig. 1: Example of BabelNet synset and related synset senses

In addition, we can enrich the nodes of the graph W_α with additional information obtained from the multilingual semantic network BabelNet [6]. BabelNet results from the automatic integration of lexicographic and encyclopaedic knowledge from WordNet senses and Wikipedia pages. Each node n_α is described by a set of BabelNet *synsets*, each synset having one or more *senses*. The set $S(n_\alpha)$ represents all senses associated to n_α. Moreover, each synset has a set of related synsets. $R(n_\alpha)$ denotes all senses belonging to the related synsets for node n_α. We note that theses senses are multilingual.

Figure 1 shows an example of BabelNet synset senses we obtain for the French Wikipedia article *Lait*. In this case only one synset *milk* is returned by BabelNet with all its senses in multiple languages such as *leche de vaca* in Spanish (ES), and *leite de vaca* in Portuguese (PT). *evaporated milk, protein, chocolate milk, white, liquid, vitamin* are examples of related synsets and their multilingual senses. For the English Wikipedia article *Coffee percolator*, two synsets *percolator* and *coffee* are returned by BabelNet where *tea*, *maxwell*, *water* and *italy* are examples of related synsets.

4 *EurekaCL* Algorithm

EurekaCL is designed to optimize the candidate selection process for any language version pair. This selection exploits the extensive linguistic information found in BabelNet to retrieve the different senses associated to a Wikipedia article source and to reach the Wikipedia articles having similar senses in any language version. For greater efficiency, the selection is then followed by a pruning procedure to eliminate the wrong candidates by examining the available cross-link paths. Finally, a ranking procedure is applied to determine the best candidate for a Wikipedia source article. The ranking procedure also uses the cross-link paths to find the most probable target article in the candidate set.

4.1 Candidate Selection

As shown in Figure 2, *EurekaCL* builds the set of BabelNet synset senses $S(n_\alpha)$ for a given Wikipedia article source n_α. Then it selects the synset senses $S^\beta(n_\alpha)$

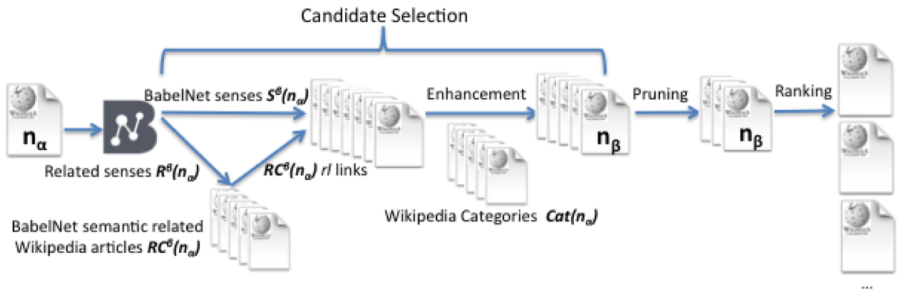

Fig. 2: *EurekaCL* algorithm

that are nouns and that represent Wikipedia articles in the target language version β to determine the set of candidate articles $C^{\beta}(n_\alpha)$.

If $C^{\beta}(n_\alpha) = \emptyset$, *EurekaCL* exploits the semantically related synset senses $R^{\beta}(n_\alpha)$ in the target language version β in order to retrieve the set $RC^{\beta}(n_\alpha)$ of Wikipedia articles semantically related to the source article n_α.

Based on the hypothesis that two Wikipedia articles covering the same topic share at least one semantically related Wikipedia article, *EurekaCL* adds to the candidate set $C^{\beta}(n_\alpha)$ all Wikipedia articles that have a symmetric intra-language link to the articles in $RC^{\beta}(n_\alpha)$. Formally:

$$\forall m_\beta \in RC^{\beta}(n_\alpha) \text{ if } \exists rl(m_\beta, n_\beta) \text{ and } \exists rl(n_\beta, m_\beta) \text{ then } n_\beta \in C^{\beta}(n_\alpha)$$

Enhancement. We define two main types of enhancements, one based on Wikipedia categories, and another based on the paths of cross-language links between the source article and the articles in the target language.

The first enhancement is meant to be an optimization of the candidate set, as it keeps only the candidates that share Wikipedia categories with the source article, based on the assumption that two Wikipedia articles covering the same topic share at least one Wikipedia category. Formally:

$$\forall c_\beta \in Cat(n_\beta), c_\alpha \in Cat(n_\alpha) \text{ if } \exists cl(c_\alpha, c_\beta), \text{ then } n_\beta \in C^{\beta}(n_\alpha)$$

We note that this enhancement is applied only if it does not reduce the set of candidates below a fixed size threshold t_{min} ($|C^{\beta}(n_\alpha)| > t_{min}$).

The second enhancement adds to the candidate set the articles of the target Wikipedia that are connected to the source article via a path of cross-links. Formally:

$$\text{if } \exists n_\beta \notin C^{\beta}(n_\alpha) \text{ and } path(n_\alpha, n_\beta) \neq \emptyset \text{ then } C^{\beta}(n_\alpha) \rightarrow C^{\beta}(n_\alpha) \cup \{n_\beta\}$$

Finally, *EurekaCL* adds to $C^{\beta}(n_\alpha)$ the Wikipedia article in language β whose title is the same as the title of source article n_α, if it exists. This will be useful for invariable named entities, whose name is the same across all latin-based alphabet language versions.

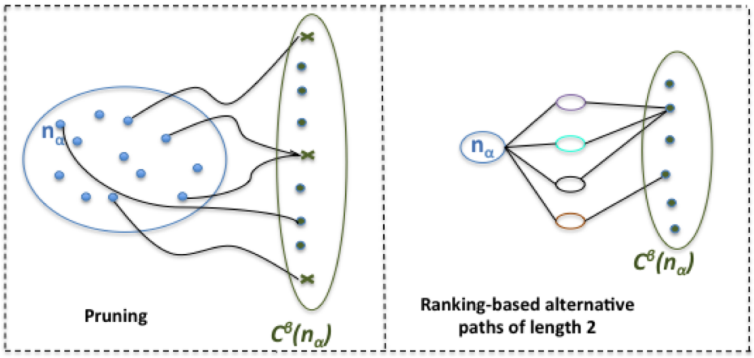

Fig. 3: Pruning and Ranking-based alternative paths illustration

4.2 Candidate Set Pruning

Once the candidate set is built, a pruning procedure is applied to remove wrong candidates. This pruning is based on the hypothesis that the cross-links are transitive. In other words, two Wikipedia articles connected by a cross-link path are likely to be about the same concept. Therefore, we define a wrong candidate as a Wikipedia article n_β not connected to the Wikipedia source article n_α but connected to another Wikipedia article m_α in language version α, as illustrated on the left of Figure 3. More precisely :

$$\exists n_\beta \in C^\beta(n_\alpha), path(n_\alpha, n_\beta) = \emptyset \text{ and } \exists m_\alpha \neq n_\alpha, path(m_\alpha, n_\beta) \neq \emptyset \text{ then}$$
$$C^\beta(n_\alpha) \rightarrow C^\beta(n_\alpha) - \{n_\beta\}$$

To retrieve all the articles in the source language version α connected to a candidate n_β, *EurekaCL* goes through the *cl* paths using a depth first search. The complexity of this search for a candidate n_β is $\mathcal{O}(i+j)$ where i and j are respectively the number of nodes and links of the *cl*-connected component including the node n_β.

4.3 Candidate Set Ranking

EurekaCL uses successively two features to rank the pruned candidate set: the *alternative paths score* then the *neighborhood links score*. To begin, we exploit the transitivity assumption to rank the target candidates.

Alternative paths score. This feature is also based on the cross-link transitivity hypothesis. Unlike the pruning procedure where the transitivity is used to eliminate candidates, at this stage *EurekaCL* profits from the fact that a source Wikipedia article is connected to a candidate Wikipedia article by a cross-link path to determine the correct target article.

However, this could lead to a conflicting situation. Indeed, as explained in Section 3 an article in a language version α may be connected by *cl* paths to more than one Wikipedia article in the target language β such that:

$$\exists n_\beta \neq m_\beta \in C^\beta(n_\alpha), \text{ such that } path(n_\alpha, n_\beta) \neq \emptyset \text{ and } path(n_\alpha, m_\beta) \neq \emptyset$$

The question then arises as to how to resolve this conflict and to choose the best target. To deal with this issue, we consider all paths connecting the source article n_α to each candidate n_β and we assume that the candidate having the highest alternative path score is the most appropriate target. We define the *alternative path score aps* for a candidate n_β as:

$$aps = \frac{|path(n_\alpha, n_\beta)|}{\sum_{m_\beta \in C^\beta(n_\alpha)} |path(n_\alpha, m_\beta)|}$$

In our evaluation, we noted that considering all paths from n_α to the candidates leads to a high computational cost. For this reason, we decided to only use the paths consisting of three nodes: n_α, one intermediate node in a certain language version and the target candidate. The situation is better explained in the right side of Figure 3. There are four paths leaving n_α, of which three lead to the upper node in the set $C^\beta(n_\alpha)$; that node is then ranked higher than the lower node.

Neighborhood links score. When the *alternative path score* does not allow any decision, *EurekaCL* exploits the neighboring articles connected to the source article and to a candidate article via *rl* intra-language links by considering both incoming and outgoing links. *EurekaCL* computes the ratio of common neighbors of each candidate $n_\beta \in C^\beta(n_\alpha)$. The common neighbor articles are retrieved thanks to the cross-links between n_α and n_β neighbors. The candidate having the largest score is ranked higher.

5 Evaluation

We downloaded from the Wikimedia Foundation website[1] the dump files of eight language versions of Wikipedia as of May 2014 and we transformed them into a Wikipedia graph, as described in Section 3, which we stored as a Neo4j 2.2.1 database. Neo4j is an increasingly popular open-source graph database that allows for easy modelling and fast traversals of large graphs, such as the Wikipedia graph on which we evaluated *EurekaCL*. To convert the Wikipedia dump files into a Neo4j database, we modified and used a tool named Graphipedia that is available on GitHub[2].

The resulting Wikipedia graph consists of 24,620,285 nodes and 329,485,368 links that are distributed among eight languages as shown in Figure 4. Most of

[1] http://dumps.wikimedia.org/backup-index.html
[2] https://github.com/gquercini/graphipedia

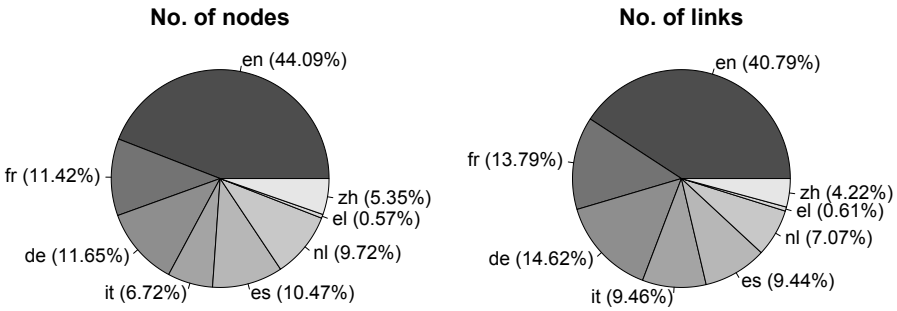

Fig. 4: The size of the Wikipedia graph

the nodes and links correspond to articles and links of the Wikipedia in English (en), which is by far the largest language version. The French (fr) and German (de) versions have a comparable size, as do the Spanish (es) and Dutch (nl) versions, at a smaller scale. The Italian (it) and Chinese (zh) versions have roughly the same amount of nodes, but considerable different links density; finally, the Wikipedia in Greek (el) includes the smallest percentage of nodes and links. We chose the languages to be evenly split between major (en, fr, es, de) and minor (it, nl, zh, el) versions, so as to assess the robustness of *EurekaCL* on graphs with a variable number of nodes and, most importantly, links.

For the evaluation of *EurekaCL*, we randomly selected a subset of 1,500 cross-links (n_α, n_β) for each language pair (α, β) in our Wikipedia graph, which resulted in a set G consisting of 42,000 cross-links (there are 28 possible language pairs). As the Wikipedia is contributed by millions of people across the world, we can assume with a certain confidence that each cross-link in G is correct, which means that it truly connects nodes corresponding to articles that cover the same concept in two different languages. Therefore, we can use G as the ground truth to evaluate *EurekaCL*.

More precisely, for each node pair $(n_\alpha, n_\beta) \in G$, *EurekaCL* is invoked to determine a *target node* m_β in the target language β for the *source node* n_α; if the output node m_β is the same as the *expected result* n_β, as per ground truth, the output of *EurekaCL* is considered to be correct for n_α. The purpose of the evaluation was to assess the number of correct answers of *EurekaCL* on all the input nodes. Obviously, we removed from the Wikipedia graph all the cross-links of G before running *EurekaCL*, and also we made sure that *EurekaCL* did not use indirectly those cross-links when querying BabelNet. To this extent, we configured *EurekaCL* so that it could not use Wikipedia as a data source when querying BabelNet for the synset senses of a source node.

All the experiments have been conducted on a computer running Linux Ubuntu 14.04 and equipped with a 8 core Intel Xeon Processor E5-2630 v3 running at 2.40GHz, 32GB of RAM and a 500GB SCSI disk.

5.1 Evaluation of the Candidate Selection

The candidate selection is a crucial step for *EurekaCL* because it can affect its output in many ways, as we will show below. In particular, we identified three properties that a good candidate selection must fulfill. First and foremost, the expected result of any source node must be included in the candidate set; if not, *EurekaCL* will not be able to output the correct node, no matter how good its ranking strategy is. Second, the size of a candidate set should be reasonably limited, otherwise the ranking step will take too long to complete. Finally, the time required to select the candidates needs to be acceptably fast.

First of all, we define the *global recall* of the candidate selection as the ratio of source nodes of G for which the expected result is included in the candidate set. We observed the global recall for different values of the size threshold t_{min}, presented in Section 4.1. If the value of t_{min} is not set, which is equivalent to saying that we do not use the Wikipedia categories, the global recall is 98%; for $t_{min} = 10$ and $t_{min} = 5$ the global recall is 95% and 93% respectively; if $t_{min} = +\infty$, which means that no threshold is set, the global recall is 88%. In our evaluation we decided to set $t_{min} = 10$ to obtain a good compromise between the global recall and the size of the candidate set, which has an impact on the computational time.

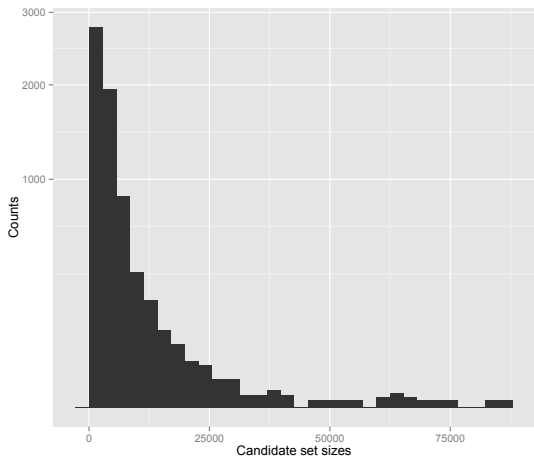

Fig. 5: The candidate set size histogram

As for the size of the candidate sets, the count histogram in Figure 5 shows that for most of the source nodes the size of the candidate set is relatively small. Indeed, for 66% of the source nodes the size is lower than 1,000, the average size is 1,372 and the maximum size is 85,154, which occurs in just one case. The first enhancement, which keeps only the candidates that share Wikipedia categories with the source article, greatly contributes to the reduction of the size of the

candidate sets. Indeed without using this enhancement at all, the average size of a candidate set would be 2,042 nodes, which corresponds to passing 28,141,309 more candidates to the ranking step .

59% of the expected results are included in the candidate set via BabelNet, while the others are included thanks to the enhancement. Among the candidates included via BabelNet only 11% are found via the synset senses, while 89% via the related synset senses. This is due to the fact that most of the source nodes in G correspond to Wikipedia articles about named entities (e.g., *Barack Obama*), for which the synset senses do not contain their translation.

Finally, the average time needed to select the candidates is 2.5 seconds; the maximum is 18 minutes and the minimum is 1ms.

5.2 Evaluation of the Candidate Pruning

Pruning proved to be very effective in reducing the number of nodes in the candidate sets. It has also a highly positive impact on the final result of the ranking, as we will show later. In Figure 6 we show the histogram of the candidate

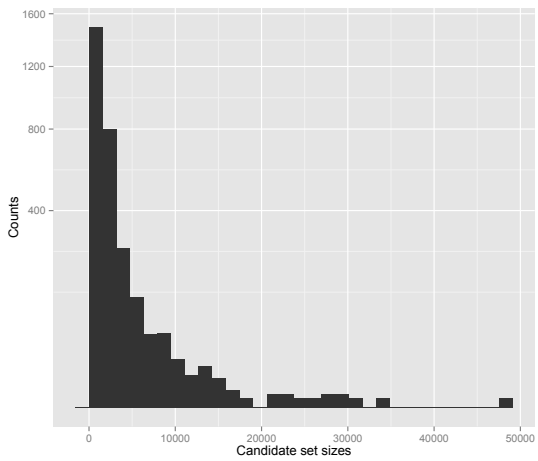

Fig. 6: The candidate set size histogram after applying pruning

set sizes; compared to the one in Figure 5 one can easily see that the average size is considerably lower than the size of the set before pruning (from 1,372 to 302); also, the maximum is now 47,545, which is almost a half of the maximum value before pruning.

As pruning involves deleting some candidates, there are chances that the expected result might be removed from the candidate set, although the pruning procedure has been conceived to remove only those candidates that are known to be connected by a path of cross-links to nodes other than the source node.

We found that for 15 source nodes, a very small percentage of our entire dataset, the expected result was removed. It is interesting to note that in these cases the problem is that some cross-links in our Wikipedia graph are not correct. One such case is the source English article titled "Flight planning" whose expected result in the German Wikipedia is "Flugplanung"; when pruning, *EurekaCL* finds that "Flugplanung" is linked by transitivity through the Spanish article "Plan de Vuelo" to the English article "Flight Plan"; since this article is different than the source article, "Flugplanung" is then pruned from the candidate set.

Finally, we point out that the average pruning time is 710ms, the maximum is 4 minutes and the minimum is 1ms.

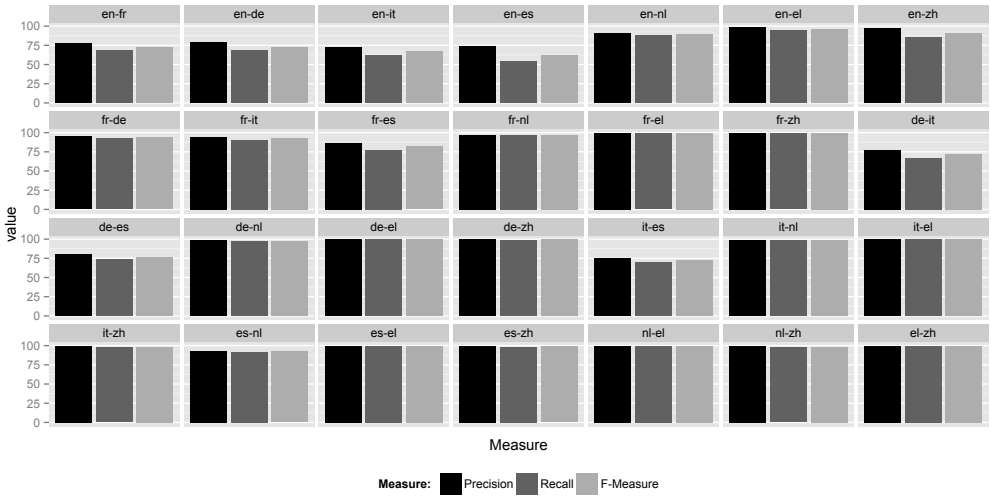

Fig. 7: Results for the Top-1 ranking by language pair

5.3 Evaluation of Ranking

For a given source node n_α, the output of *EurekaCL* is a list of nodes sorted by decreasing score; ideally, the top node in the list corresponds to the expected result n_β. In this section, we evaluate EurekaCL's ranking process. We look at its ability to rank the expected result first or among the top 10 candidates. To this extent, we use three measures – *precision*, *recall*, and *f-measure* – that we define as follows. Let G be the ground truth as previously defined and T_1, T_3, T_5, and T_{10} the sets of source nodes for which the expected result is ranked first, in the top 3, top 5, and top 10 respectively. Also, let I be the set of source nodes for which the expected result is included in the candidate set. We define a set of measures of precision P_i, recall R_i, f-measure F_i, for $i = 1, 3, 5, 10$ as follows:

$$P_i = \frac{|T_i|}{|I|} * 100 \quad R_i = \frac{|T_i|}{|G|} * 100 \quad F_i = \frac{2 * P_i * R_i}{P_i + R_i}$$

The results of the ranking for the Top-1 case for each language pair are shown in Figure 7. First of all, we note that the average precision P_1, recall R_1 and f-measure F_1 across all language pairs are 94.24%, 91.29% and 92.68% respectively. Based on the figure, it is evident that for some language pairs the results are sensibly lower than the average. This is especially true for the pairs en-fr, en-de, en-it and en-es. The problem here is that in most (1,248 out of 1,500) of the pairs $(n_\alpha, n_\beta) \in G$, such that α is the English language and β is the French language, the source node n_α is isolated, in that there is no cross-links incident with it (except the one that connects it to n_β, which has been removed for the evaluation). As a result, the ranking for these source nodes is based solely on the neighborhood score, while the alternative paths score never kicks in.

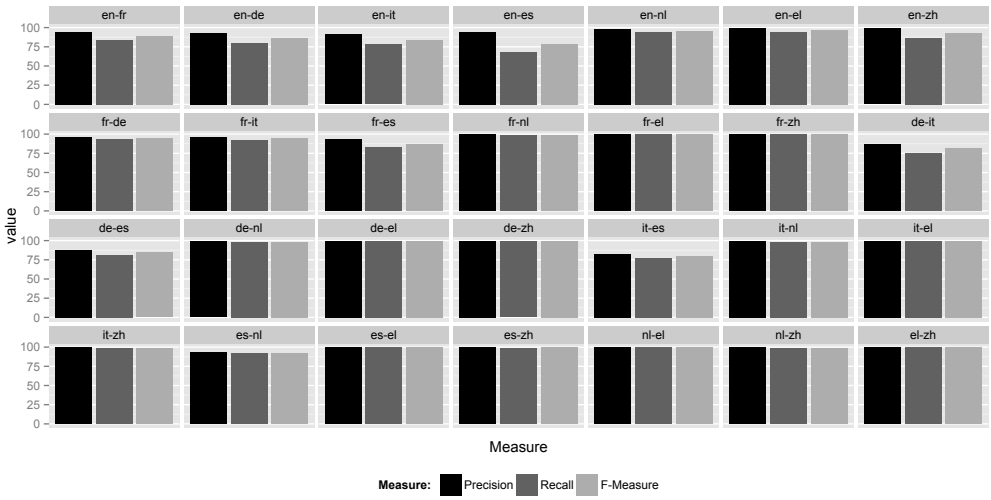

Fig. 8: Results for the Top-1 ranking by language pair after pruning

The benefits of the pruning on the ranking are shown in Figure 8. We note that the precision/recall improve considerably, especially for those language pairs that we pointed out above. For the en-fr pair, for instance, the precision jumps from 77% to 94% and the recall from 68.93% to 84.20%. The average precision P_1, recall R_1 and f-measure F_1 across all language pairs are 97.04%, 93.73% and 95.28% respectively.

Table 1 shows a summary of the results of the ranking. The value between parentheses in each cell refers to the result obtained after pruning.

		Top-1	Top-3	Top-5	Top-10
Precision	**no pruning**	94.24	97.64	98.84	99.80
	pruning	97.04	98.83	99.34	99.62
Recall	**no pruning**	91.29	94.33	95.41	96.29
	pruning	93.73	95.36	95.83	96.10
F-measure	**no pruning**	92.68	95.88	97.02	97.93
	pruning	95.28	96.98	97.47	97.75

Table 1: Results of the ranking

5.4 Comparison

We compare *EurekaCL* against two existing approaches, namely *WikiCL* [7] and Sorg&Cimiano [8], which have been previously evaluated on a publicly available dataset. This dataset, referred to as RAND1000, contains 1000 pairs of articles such that the source article is in the English Wikipedia and the expected result is in the German Wikipedia.

	Precision	Recall	F-measure
EurekaCL	**99.14**	**97.78**	**98.46**
WikiCL	89.00	88.00	89.00
Sorg&Cimiano	94.00	70.00	80.00

Table 2: Comparison

The results of the comparison, summarized in Table 2, show that *EurekaCL* clearly outperforms the other approaches, on both precision and recall. We note that the global recall for the candidate selection of *EurekaCL* is 98.62%, while in Sorg&Cimiano is 86.5%; the average candidate set size for *EurekaCL* is 753, while for Sorg&Cimiano is 1,000. These values are not available for *WikiCL*.

6 Conclusion

In this paper we presented and evaluated a multilingual approach called *EurekaCL* to automatically identify missing cross-language links in Wikipedia. The strongest points of *EurekaCL* are an effective candidate selection and pruning strategy, which reduces the number of candidates to rank and considerably improves the final precision and recall, as our experiments showed. Our evaluation is carried out on a large dataset, including 4 major and 4 minor Wikipedia

language versions. We also ran *EurekaCL* on RAND1000, a publicly available dataset, to make a comparison with two existing approaches.

As future work, our goal is threefold. First of all, we intend to optimize the implementation of *EurekaCL* and the Wikipedia graph representation to speed up the ranking, which is the step that takes most of the time compared to the candidate selection step (9 seconds on average, with a maximum value of 45 minutes). Next, we plan to generalize *EurekaCL* to detect missing cross-links between Wikipedia pages belonging to other namespaces, in particular categories. Finally, we will also focus our attention on the problem of detecting erroneous cross-language links in Wikipedia, as done in [4].

References

1. S. F. Adafre and M. de Rijke. Finding Similar Sentences across Multiple Languages in Wikipedia. *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 62–69, 2006.
2. Alessio Palmero Aprosio, Claudio Giuliano, and Alberto Lavelli. In Philipp Cimiano, Óscar Corcho, Valentina Presutti, Laura Hollink, and Sebastian Rudolph, editors, *ESWC*, volume 7882 of *Lecture Notes in Computer Science*, pages 397–411. Springer, 2013.
3. Gerard de Melo and Gerhard Weikum. Menta: inducing multilingual taxonomies from wikipedia. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 1099–1108. ACM, 2010.
4. Gerard de Melo and Gerhard Weikum. Untangling the cross-lingual link structure of wikipedia. In *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*, pages 844–853. Association for Computational Linguistics, 2010.
5. Carlos Eduardo M. Moreira and Viviane Pereira Moreira. Finding missing cross-language links in wikipedia. *JIDM Journal of Information and Data Management*, 4(3):251–265, 2013.
6. Roberto Navigli. Babelnet and friends: A manifesto for multilingual semantic processing. *Intelligenza Artificiale*, 7(2):165–181, 2013.
7. Antonio Penta, Gianluca Quercini, Chantal Reynaud, and Nigel Shadbolt. Discovering cross-language links in wikipedia through semantic relatedness. In *ECAI 2012 - 20th European Conference on Artificial Intelligence*, pages 642–647, 2012.
8. Philipp Sorg and Philipp Cimiano. Enriching the crosslingual link structure of wikipedia -a classification-based approach-. In *Proceedings of the AAAI 2008 Workshop on Wikipedia and Artificial Intelligence (WikiAI'08)*, To appear, 2008.
9. Philipp Sorg and Philipp Cimiano. Exploiting wikipedia for cross-lingual and multilingual information retrieval. *Data Knowl. Eng.*, 74:26–45, 2012.
10. Takashi Tsunakawa, Makoto Araya, and Hiroyuki Kaji. Enriching wikipedia's intra-language links by their cross-language transfer. In *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference*, pages 1260–1268, 2014.