An Evaluation of the Effect of Reference Strings and Segmentation on Citation Matching

Behnam Ghavimi, Wolfgang Otto, and Philipp Mayr

GESIS – Leibniz Institute for the Social Sciences
firstname.lastname@gesis.org

Abstract. In this paper, three different possible inputs (reference strings, reference segments and a combination of reference strings and segments) were tested to find the best performing strategy for citation matching. Our evaluation on a manually curated gold standard showed that the input data consisting of the combination of reference segments and reference strings lead to the best result. In addition, the usage of the probabilities of the segmentation improve the result when only features based on reference segments are considered.

Keywords: Citation Matching · Reference Segments · Reference Strings · Evaluation

1 Introduction

The process of mapping an extracted reference string from a publication to one entity of a given Digital Library (DL) is called citation matching. Proper citation matching is an essential step for every citation analysis [6] and the improvement of citation matching leads to a higher quality of bibliometric studies. Christen et al. [2] reviewed different matching approaches and based on their study, they suggested general steps for the record linkage process: input pre-processing, blocking technique, feature extraction and classification. We also followed these steps for citation matching in this experiment and consider different input configurations to investigate their effects. In the pre-processing step the reference strings are segmented. Our used segmentation tool was Exparser [1] which includes a probability for each predicted segment. The next step is the blocking technique in order to decrease the number of pairs required to be compared. We used the search engine Solr for blocking. Koo et al. tried to find the best combination of citation record fields [5] that helps increase citation matching performance. We used a similar approach to shrink the number of queries in the blocking step. In the final step of the citation matching process, each candidate record pair (i.e, reference string and related segments paired with each retrieved item of the DL) are compared using a variety of attributes and comparison functions to generate a vector of features. Each candidate record pair is classified into one of the classes *match* and *non-match* based on their vector of features. Wellner et al. [7] investigated the effect of extraction probabilities on citation matching by the

consideration of more than one of the top segmentation sequence probabilities based on the Viterbi algorithm. Our approach considers only the most probable segmentation. The probability of each segment is used in additional features that are utilized by the binary classifier for the citation matching task.

2 Evaluation Setup

For each reference, our matching algorithm[1] in this experiment retrieves the corresponding block with the help of blocking queries. Queries are formulated with the help of the parsed segments and reference strings by using the operators OR and AND from the Solr query syntax[2]. Additional we use fuzzy search ($\sim$-operator) which reflects a fuzzy string similarity search based on the Levenshtein distance. The threshold for this score was defined as 0.7 experimentally. The output of the blocking step is a ranked lists of retrieved items from the target database. The items are ranked by the tf/idf[3]-based Lucene score. To get the best trade off between retrieving all possible matching items and the reduction of necessary comparisons in the following classification task we identified two opportunities for influence. One is varying the query and select the best query formulation. The other is the selection of a cut off threshold which determines how many of the retrieved items per query are used for further processing. To exclude not well performing segment combinations for query generation we measured the 'precision@1' of the queries on our gold data. We only select segment combinations where at least 60% of the retrieved items are a correct match. This reduces the number of maximum combinations we consider for query generation by 25%. As an alternative strategy we generated queries from the reference strings. For this we consider all token of the reference string as potentially including title information. The idea is to formulate a bigram search of the whole reference string. The resulting query leads to results which at least need to include one bigram of the reference string in the title field. But the more bigrams of the reference string are included in the title, the more preferred results are. In addition, to increase the precision, a query based on year and bigrams of the reference string will also be considered. For this, the year information is taken into account which is extracted with a regular expression. After retrieving candidates for matches with our blocking procedure we need to decide which of the found candidates our system identifies as a match. We used features generated from the raw reference string and from the segmentation. Also we tried to find the effect of utilizing the certainty of our parser for the detected segments as an additional input feature for our classifier. The first group of features is based on the comparison of the reference segments and the retrieved items in the blocking step: 1. Features based on the author segment (e.g., A. Levenshtein score (phonocode and exact) and B. Segmentation probability of first author (surname)), 2.

[1] `https://github.com/exciteproject/EXmatcher`

[2] `https://lucene.apache.org/solr/guide/6_6/query-syntax-and-parsing.html`

[3] `https://lucene.apache.org/core/7_4_0/core/org/apache/lucene/search/similarities/\TFIDFSimilarity.html`

Features based on titles (e.g., A. Jaccard score (including segmentation probabilities) and B. Levenshtein score (token and character level)), and some other feature which for reasons of space are not listed here. The second feature group are based on the comparison of the raw reference string with the information of the retrieved record. Examples features for this group are the longest common sub-string of title and reference string, and the occurrence of the abbreviation of the source field (e.g., journal abbreviation) of retrieved item in the reference string.

3 Evaluation

The matching target DL used in this paper is Sowiport [3] which contains bibliographic metadata records of more than 9 million references. A manually checked gold standard was generated for this evaluation. This corpus include the information about 816 reference strings (randomly selected from EXCITE corpus which contains about 300K publications in PDF format) and all their corresponding items in Sowiport. 517 of these items have at least one matched item. We published this corpus and a part of sowiport data (18,590 bibliographic items) openly in our Github repository[4]. Three different configurations for the input of blocking were examined. In addition, the effect of the consideration of different numbers of top items from the blocking step was checked. The result shows that the precision of blocking based on reference strings is higher than the two other configurations. The highest recall has been achieved using the combination of reference strings and segments. The consideration of more top items shrinks the recall gap between different input configurations. Since we have another step after blocking which improve the precision, the important point in blocking is to keep the recall score high and at the same time shrinking the number of items for comparison. The precision of these three curves were not significantly different, therefore, the combination of reference strings and segments is picked in blocking step to generate input for the evaluation of classification step. For the number of top items in blocking, which are used for further processing in our experience, five is selected. This value is chosen because considering more items doesn't have a significant improvement on the recall score of using the combination of reference strings and segments as input but it generates much more false pairs. For the 816 references of the gold standard 10,997 match candidates are generated with our selected configuration in blocking. These candidates are based on top 5 retrieved items of all considered queries for each reference. In these 10,997 pairs, 1,026 (9.3%) are correct matches and 9,971 (90.7%) are no matches. After blocking, the number of reference strings which have at least one correct match is 507, and 302 references are without any correct pair. It means, for only ten references (1.2%) which have at least one match in the gold standard could not pass blocking successfully. We applied ten-fold cross validation for testing two classifiers (SVM and Random Forest) and feature combinations. Table 1 contains precision, recall and f-measure for our compared configurations.

[4] `https://github.com/exciteproject/EXgoldstandard/`

Ref_String	Ref_Segments	Seg_probability	SVM	Random Forest	Precision	Recall	F1
✓	✓	✓	✓	-	0.947 *	0.904	0.925 *
✓	✓	✓	-	✓	0.938	0.906	0.921
✓	✓	-	✓	-	0.941	0.908 *	0.924
✓	✓	-	-	✓	0.923	0.899	0.910
-	✓	✓	✓	-	0.942	0.865	0.901
-	✓	✓	-	✓	0.918	0.874	0.895
-	✓	-	✓	-	0.836	0.869	0.852
-	✓	-	-	✓	0.876	0.883	0.879
✓	-	-	✓	-	0.843	0.903	0.871
✓	-	-	-	✓	0.879	0.855	0.866

Table 1. Evaluation macro-metrics of different classifiers.

4 Discussion and Conclusions

We analyzed the impact of different inputs (i.e., reference strings, segments and the combination of both) on citation matching procedure. The segmentation probabilities are considered directly and as weights for creating specific features for classifiers. Segments probabilities have a good impact on the precision score when the citation matching algorithm uses segments as input. Using the combination of reference strings and segments as input outperforms the other configurations. The effect of different classifiers on the result are very depended on other parameters in the citation matching configuration such as input types and the consideration of segment probabilities. The citation matching approach which has been described and evaluated in this paper is implemented in a demonstrator which connects all important steps from reference extraction, reference segmentation and matching in the EXCITE toolchain (see [4] `http://excite.west.uni-koblenz.de/excite`).

References

1. Boukhers, Z., et al.: An End-to-end Approach for Extracting and Segmenting High-Variance References from PDF Documents. In: Proc. of JCDL 2019. ACM (2019)
2. Christen, P.: Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection. Springer Science & Business Media (2012)
3. Hienert, D., et al.: Digital Library Research in Action - Supporting Information Retrieval in Sowiport. D-Lib Magazine **21**(3/4) (2015)
4. Hosseini, A., et al.: EXCITE - A toolchain to extract, match and publish open literature references. In: Proc. of JCDL 2019 (2019)
5. Koo, H.K., et al.: Effects of unpopular citation fields in citation matching performance. In: Proc. of ICISA 2011 (2011)
6. Moed, H.F.: Citation analysis in research evaluation, vol. 9. Springer Science & Business Media (2006)
7. Wellner, B., et al.: An integrated, conditional model of information extraction and coreference with application to citation matching. AUAI Press (2004)

View publication stats