Off-the-shelf Semantic Author Name Disambiguation for Bibliographic Data Bases

Mark-Christoph Müller[1(✉)], Adam Bannister[2], and Florian Reitz[3]

[1] Heidelberg Institute for Theoretical Studies gGmbH, Heidelberg, Germany
mark-christoph.mueller@h-its.org
[2] Mathematics Department, FIZ Karlsruhe, Berlin, Germany
adam.bannister@fiz-karlsruhe.de
[3] Schloss Dagstuhl LZI, Wadern, Germany
florian.reitz@dagstuhl.de

Abstract. The demo presents a minimalist, off-the-shelf AND tool which provides a fundamental AND operation, the comparison of two publications with ambiguous authors, as an easily accessible HTTP interface. The tool implements this operation using standard AND functionality, but puts particular emphasis on advanced methods from natural language processing (NLP) for comparing publication title *semantics*.

Keywords: Author name disambiguation · Semantic similarity · Word embeddings · API · Open source software

1 Introduction and Motivation

Institutions where bibliographic data is collected, processed, and stored on a large scale – like e.g. digital libraries – frequently encounter the **author name ambiguity** problem: two or more identical, or highly similar, author ***names*** appear in the headers of different publications, but it is uncertain whether these names refer to the same author ***individual***. Author name ambiguity mainly results from a combination of the following: 1. very common names, 2. publishers' practice of abbreviating first names, and 3. lack of consistency on the part of the authors [9].

Author name disambiguation (AND) attempts to resolve the referential uncertainty of author names by automatically distinguishing them on the basis of a wide range of properties, and assigning to them a collection-wide unique identifier.[1] The difficulty of correctly disambiguating two ambiguous author names in two publications ranges from *trivial* to *virtually impossible*, and mainly depends on the following factors: – the availability of general **author and publication**

[1] In contrast, the non-technical, *organizational* approaches of orcid.org or researcherid.com attempt to *prevent* referential uncertainty, by having authors use globally unique identifiers in their publications.

A. Doucet et al. (Eds.): TPDL 2019, LNCS 11799, pp. 397–400, 2019.
https://doi.org/10.1007/978-3-030-30760-8_42

meta data, e.g. complete author names, email addresses, affiliations, and publication venues; – the type of publication, e.g. **single- or multi-author**; and – the degree of specialization of the publication **topic**, normally observable in its **title**. At one extreme end of the spectrum, both author names are accompanied by matching email addresses, which are almost perfect author identifiers. At the other extreme, each of the two publications features a run-off-the-mill title and a single author with a very common name[2].

In this demo, we present our open-source Python implementation of a simple, lightweight, and extensible AND tool. Its functionality – which currently consists of one elementary AND operation – is exposed via an HTTP API and can be used in isolation (e.g. via a web browser), or as the basis for implementing higher-level AND workflows in practically every modern programming language. Due to the tool's minimalist approach, it is runnable off-the-shelf, i.e. without extensive configuration, let alone training. A specific back-end data base is not required, either, since all author and publication meta data needed for disambiguation are provided by the user in the API function call. The tool and pre-trained resources are available and will continue to be maintained at github.com/nlpAThits/scad-tool.

In recent years, many different AND systems have been proposed and published (cf. [1,2]), but as far as we can see none of them has been accepted as a standard or *best practice* by the community. One problem is that existing AND systems implement the task in different ways, e.g. incrementally vs. non-incrementally, record- vs. profile-based, grouping- vs. assignment-based [1], or online (i.e. processing *one new* record at a time) vs. batch (i.e. processing *a whole block* of records at once) [4]. Also, systems are often solely applied to, and sometimes even tailored towards, particular bibliographic data bases (like e.g. PubMed, MEDLINE, CiteSeerX, or dblp), or they are tested and optimized on particular AND test collections (cf. Müller et al. [9] for an overview), which also limits their broader applicability. Another problem is that in many cases, complete, well-maintained source code is either not available at all, or apparently outdated, as Kim et al. [5] observe with CiteSeerX[3] and AMiner[4]. However, Kim et al. do not provide source code for their system, either. In contrast, our tool is completely agnostic to particular data bases, research disciplines, and AND workflow implementations, and the source code is freely available.

Many existing AND systems strongly rely on **coauthor information** for disambiguation, which is a reasonable strategy in those research disciplines where publications commonly have multiple authors. Actually, the popularity of coauthor-based AND and the high prevalence of multi-author publications in AND test collections [9] can be seen as mutually affecting each other. However, there are also many research disciplines in the real world where author collaboration is much less common, and for which most existing AND systems will fail to produce acceptable results.

[2] See e.g. dblp.uni-trier.de/pers/hd/w/Wang:Wei.
[3] github.com/SeerLabs/CiteSeerX.
[4] github.com/askerlee/namedis.

Semantic similarity between two publications, on the other hand, is a domain-independent potential indicator for author identity, whose usefulness has been demonstrated already [7]. However, with only a few exceptions (e.g. [5–7]), the majority of currently existing AND systems recognizes similarity between two publications' titles, keywords, or abstracts on the *surface* level only, i.e. by simple string matching over lists of white-space-separated tokens, word stems, or character n-grams. In natural language processing (NLP), **word embeddings** are now the generally accepted standard method for quantifying semantic similarity beyond the string level.[5] Since word embeddings can be trained with comparably little effort on large collections of raw text, they can be employed as resources for computing **domain-specific semantic similarity**. Our tool supports this flexible use of different word embeddings by accepting word embedding identifiers as *parameters* in the API function call.

2 `match_authors` as an Atomic AND Procedure

Our tool currently provides the atomic procedure `match_authors`, which analyses the meta data of two publications with ambiguously named authors, and returns *True* if it classifies the authors as identical, and *False* otherwise. The classification is accompanied by a confidence score. The procedure is similar to the 'record-based query' of Kim et al. [5], but with the important difference that `match_authors` expects *both* publications to be provided by the user, while the system of Kim et al. tries to match *one* user-provided publication against pre-existing publications in its back-end data base. The API expects the meta data for the two publications as one JSON object each. We use a simple and straightforward JSON format which can be easily extended, e.g. to cover publications from sources which provide richer meta data. The following is an example of a publication from the KISTI data set [3], which is based on data from dblp.org.

```
{'id':      'dblp:journals/taslp/KarmakarKP06',
 'title':   'A Multiresolution Model of Auditory Excitation Pattern and
             Its Application to Objective Evaluation of Perceived Speech Quality',
 'authors': [
            {'name': 'A. Karmakar',  'shortname': 'A. Karmakar'},
            {'name': 'A. Kumar',     'shortname': 'A. Kumar', 'id': 'A.Kumar_8'},
            {'name': 'R. K. Patney', 'shortname': 'R. Patney'}],
 'year':    2006,
 'venue': 'journals/taslp',
 'pages': '1912-1923',
 'classifications': {} }
```

The second example comes from the SCAD-zbMATH AND data set [9], which is based on data from zbmath.org. This publication features some additional meta data, incl. keywords, which are particularly relevant for semantic AND.

```
{'id':     'zbmath:0614.93069',
 'title':  'Positional modeling of stochastic control in dynamical systems',
 'authors': [
            {'name': 'Osipov, Yu.S.',      'shortname': 'Osipov, Yu.',      'id': 'osipov.yuri-s'},
            {'name': 'Kryazhimskij, A.V.', 'shortname': 'Kryazhimskij, A.', 'id': 'kryazhimskii.arkadii-v'}],
 'year': 1986,
 'venue': 'Stochastic optimization, Proc. Int. Conf., Kiev/USSR 1984, Lect. Notes Control Inf. Sci. 81, 696-704 (1986).',
 'classifications': {
                  'msc-codes': ['93E20', '34A55', '93C10', '91A23', '93C15'],
                  'keywords': ['inverse dynamical problems', 'stochastic controls']}}
```

[5] E.g. github.com/tmikolov/word2vec, github.com/facebookresearch/fastText, github.com/allenai/allennlp/blob/master/tutorials/how_to/elmo.md.

`match_authors` is called with two publications' JSON strings and the ambiguous author position in each (as an index into the authors JSON array). In addition, it can accept a word embedding identifier in WOMBAT format [10]. WOMBAT is used for efficient word-level retrieval of vector representations, which are the main input for computing semantic similarity scores. The use of WOMBAT allows the system to dynamically select a word embedding resource for a particular domain (e.g. computer science, math, chemistry, etc.) when publications from a corresponding venue are processed. In order to increase the transparency and acceptability of the automatic classification, semantic similarity scores are computed in such a way that they yield both a numerical value and a compact, human-interpretable representation of what exactly went into the computation [8]. This way, sanity checking by a human expert is easily implemented.

Since the whole design of our tool is open and extensible, the implementation details can change as long as the method signature and its in- and output requirements are observed.

Acknowledgements. The work described in this paper was conducted in the project *SCAD – Scalable Author Disambiguation*, funded in part by the Leibniz Association (grant SAW-2015-LZI-2), and in part by the Klaus Tschira Foundation.

References

1. Ferreira, A.A., Gonçalves, M.A., Laender, A.H.: A brief survey of automatic methods for author name disambiguation. SIGMOD Rec. **41**(2), 15–26 (2012)
2. Hussain, I., Asghar, S.: A survey of author name disambiguation techniques: 2010–2016. Knowl. Eng. Rev. **32**, 1–24 (2017)
3. Kang, I.S., Kim, P., Lee, S., Jung, H., You, B.J.: Construction of a large-scale test set for author disambiguation. Inf. Process. Manag. **47**(3), 452–465 (2011)
4. Khabsa, M., Treeratpituk, P., Giles, C.L.: Large scale author name disambiguation in digital libraries. In: BigData, pp. 41–42. IEEE Computer Society (2014)
5. Kim, K., Sefid, A., Weinberg, B.A., Giles, C.L.: A web service for author name disambiguation in scholarly databases. In: ICWS, pp. 265–273. IEEE (2018)
6. Müller, M.-C.: Semantic author name disambiguation with word embeddings. In: TPDL, pp. 300–311 (2017). https://doi.org/10.1007/978-3-319-67008-9_24
7. Müller, M.-C.: On the contribution of word-level semantics to practical author name disambiguation. In: JCDL, pp. 367–368. ACM (2018). https://doi.org/10.1145/3197026.3203912
8. Müller, M.-C.: Semantic matching of documents from heterogeneous collections: a simple and transparent method for practical applications. In: RELATIONS, pp. 34–41 (2019). https://www.aclweb.org/anthology/W19-0804
9. Müller, M.-C., Reitz, F., Roy, N.: Data sets for author name disambiguation: an empirical analysis and a new resource. Scientometrics **111**(3), 1467–1500 (2017). https://doi.org/10.1007/s11192-017-2363-5
10. Müller, M.-C., Strube, M.: Transparent, efficient, and robust word embedding access with WOMBAT. In: COLING (System Demonstrations), pp. 53–57 (2018). https://aclweb.org/anthology/papers/C/C18/C18-2012