OpenTapioca: Lightweight Entity Linking for Wikidata

Antonin Delpeuch[1] [0000−0002−8612−8827]

Department of Computer Science, University of Oxford, UK
antonin.delpeuch@cs.ox.ac.uk

Abstract. We propose a simple Named Entity Linking system that can be trained from Wikidata only. This demonstrates the strengths and weaknesses of this data source for this task and provides an easily reproducible baseline to compare other systems against. Our model is lightweight to train, to run and to keep synchronous with Wikidata in real time.

Keywords: Entity linking · Wikidata

1 Introduction

Named Entity Linking is the task of detecting mentions of entities from a knowledge base in free text, as illustrated in Figure 1.

Most of the entity linking literature focuses on target knowledge bases which are derived from Wikipedia, such as DBpedia [1] or YAGO [19]. These bases are curated automatically by harvesting information from the info-boxes and categories on each Wikipedia page and are therefore not editable directly.

Wikidata [21] is an editable, multilingual knowledge base which has recently gained popularity as a target database for entity linking [8,22,15,13]. As these new approaches to entity linking also introduce novel learning methods, it is hard to tell apart the benefits that come from the new models and those which come from the choice of knowledge base and the quality of its data.

We review the main differences between Wikidata and static knowledge bases extracted from Wikipedia, and analyze their implactions for entity linking. We illustrate these differences by building a simple entity linker, OpenTapioca[1], which only uses data from Wikidata, and show that it is competitive with other systems with access to larger data sources for some tasks. OpenTapioca can be trained easily from a Wikidata dump only, and can be efficiently kept up to date in real time as Wikidata evolves. We also propose tools to adapt existing entity linking datasets to Wikidata, and offer a new entity linking dataset, consisting of affiliation strings extracted from research articles.

[1] The implementation and datasets are available at https://github.com/wetneb/opentapioca and the demo can be found at https://opentapioca.org/.

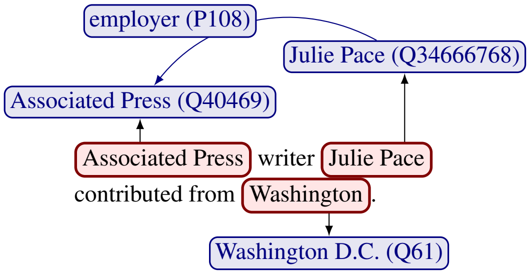

Fig. 1. Example of an annotated sentence with mentions in red and entities in blue

2 Particularities of Wikidata

Wikidata is a wiki itself, meaning that it can be edited by anyone, but differs from usual wikis by its data model: information about an entity can only be input as structured data, in a format that is similar to RDF.

Wikidata stores information about the world in a collection of *items*, which are structured wiki pages. Items are identified by ther Q-id, such as Q40469, and they are made of several data fields. The *label* stores the preferred name for the entity. It is supported by a *description*, a short phrase describing the item to disambiguate it from namesakes, and *aliases* are alternate names for the entity. These three fields are stored separately for each language supported by Wikidata. Items also hold a collection of *statements*: these are RDF-style claims which have the item as subject. They can be backed by *references* and be made more precise with *qualifiers*, which all rely on a controlled vocabulary of *properties* (similar to RDF predicates). Finally, items can have *site links*, connecting them to the corresponding page for the entity in other Wikimedia projects (such as Wikipedia). Note that Wikidata items do not need to be associated with any Wikipedia page: in fact, Wikidata's policy on the notability of the subjects it covers is much more permissive than in Wikipedia. For a more detailed introduction to Wikidata's data model we refer the reader to [21,5].

Our goal is to evaluate the usefulness of this crowdsourced structured data for entity linking. We will therefore refrain from augmenting it with any external data (such as phrases and topical information extracted from Wikipedia pages), as is generally done when working with DBpedia or YAGO. By avoiding a complex mash-up of data coming from disparate sources, our entity linking system is also simpler and easier to reproduce. Finally, it is possible keep OpenTapioca in real-time synchronization with the live version of Wikidata, with a lag of a few seconds only. This means that users are able to fix or improve the knowledge base, for instance by adding a missing alias on an item, and immediately see the benefits on their entity linking task. This constrasts with all other systems we are aware of, where the user either cannot directly intervene on the underlying data, or there is a significant delay in propagating these updates to the entity linking system.

3 Related work

We review the dominant architecture of entity linking heuristics following [?], and assess its applicability to Wikidata.

Entities in the knowledge base are associated with a set (or probability distribution) of possible surface forms. Given a text to annotate, candidate entities are generated by looking for occurrences of their surface forms in the text. Because of homonymy, many of these candidate occurrences turn out to be false matches, so a classifier is used to predict their correctness. We can group the features they tend to use in the following categories:

- **local compatibility**: these features assess the adequacy between an entity and the phrase that refers to it. This relies on the dictionary of surface forms mentioned above, and does not take into account the broader context of the phrase to link.
- **topic similarity**: this measures the compatibility between the topics in the text to annotate and the topics associated with the candidate entity. Topics can be represented in various ways, for instance with a bag of words model.
- **mapping coherence**: entities mentioned in the same text are often related, so linking decisions are inter-dependent. This relies on a notion of proximity between entities, which can be defined with random walks in the knowledge base for instance.

3.1 Local compatibility

These features compare the phrase to annotate with the known surface forms for the entity. Collecting such forms is often done by extracting mentions from Wikipedia [3]. Link labels, redirects, disambiguation pages and bold text in abstracts can all be useful to discover alternate names for an entity. It is also possible to crawl the web for Wikipedia links to improve the coverage, often at the expense of data quality [16].

Beyond collecting a set of possible surface forms, these approaches count the number of times an entity e was mentioned by a phrase w. This makes it possible to use a Bayesian methodology: the compatibility of a candidate entity e with a given mention w is $P(e|w) = \frac{P(e,w)}{P(w)}$, which can be estimated from the statistics collected.

In Wikidata, items have labels and aliases in multiple languages. As this information is directly curated by editors, these phrases tend to be of high quality. However, they do not come with occurence counts. As items link to each other using their Wikidata identifiers only, it is not possible to compare the number of times `USA` was used to refer United States of America (Q30) or to United States Army (Q9212) inside Wikidata.

Unlike Wikipedia's page titles which must be unique in a given language, two Wikidata items can have the same label in the same language. For instance `Curry` is the English label of both the item about the Curry programming language (Q2368856) and the item about the village in Alaska (Q5195194), and the description field is used to disambiguate them.

Manual curation of surface forms implies a fairly narrow coverage, which can be an issue for general purpose entity linking. For instance, people are commonly refered to with their given or family name only, and these names are not systematically added as aliases: at the time of writing, `Trump` is an alias for Donald Trump (Q22686), but `Cameron` is not an alias for David Cameron (Q192). As a Wikidata editor, the main incentive to add aliases to an item is to make it easier to find the item with Wikidata's auto-suggest field, so that it can be edited or linked to more easily. Aliases are not designed to offer a complete set of possible surface forms found in text: for instance, adding common mispellings of a name is discouraged.[2]

3.2 Topic similarity

The compatibility of the topic of a candidate entity with the rest of the document is traditionally estimated by similarity measures from information retrieval such as TFIDF [17,14] or keyword extraction [18,10,3].

Wikidata items only consist of structured data, except in their descriptions. This makes it difficult to compute topical information using the methods above. Vector-based representations of entities can be extracted from the knowledge base alone [2,24], but it is not clear how to compare them to topic representations for plain text, which would be computed differently. In more recent work, neural word embeddings were used to represent topical information for both text and entities [4,13,9]. This requires access to large amounts of text both to train the word vectors and to derive the entity vectors from them. These vectors have been shown to encode significant semantic information by themselves [11], so we refrain from using them in this study.

3.3 Mapping coherence

Entities mentioned in the same context are often topically related, therefore it is useful not to treat linking decisions in isolation but rather to try to maximize topical coherence in the chosen items. This is the issue on which entity linking systems differ the most as it is harder to model.

First, we need to estimate the topical coherence of a sequence of linking decisions. This is often done by first defining a pairwise relatedness score between the target entities. For instance, a popular metric introduced by [23] considers the set of wiki links $|a|, |b|$ made from or to two entities a, b and computes their relatedness:

$$\mathrm{rel}(a,b) = 1 - \frac{\log(\max(|a|,|b|)) - \log(|a \cap b|)}{\log(|K|) - \log(\min(|a|,|b|))}$$

where $|K|$ is the number of entities in the knowledge base.

When linking to Wikidata instead of Wikipedia, it is tempting to reuse these heuristics, replacing wikilinks by statements. However, Wikidata's linking structure is quite different from Wikipedia: statements are generally a lot sparser than links and they have a precise semantic meaning, as editors are restricted by the available properties when creating new statements. We propose in the next section a similarity measure that we find to perform well experimentally.

Once a notion of semantic similarity is chosen, we need to integrate it in the inference process. Most approaches build a graph of candidate entities, where edges indicate

[2] The guidelines are available at `https://www.wikidata.org/wiki/Help:Aliases`

semantic relatedness: the difference between the heuristics lie in the way this graph is used for the matching decisions. [12] use an approximate algorithm to find the *densest subgraph* of the semantic graph. This determines choices of entities for each mention. In other approaches, the initial evidence given by the local compatibility score is propagated along the edges of the semantic graph [10,6] or aggregated at a global level with a Conditional Random Field [4].

4 OpenTapioca: an entity linking model for Wikidata

We propose a model that adapts previous approaches to Wikidata. Let d be a document (a piece of text). A *spot* $s \in d$ is a pair of start and end positions in d. It defines a phrase $d[s]$, and a set of candidate entities $E[s]$: those are all Wikidata items for which $d[s]$ is a label or alias. Given two spots s, s' we denote by $|s - s'|$ the number of characters between them.[3] We build a binary classifier which predicts for each $s \in d$ and $e \in E[s]$ if s should be linked to e.

4.1 Local compatibility

Although Wikidata makes it impossible to count how often a particular label or alias is used to refer to an entity, these surface forms are carefully curated by the community. They are therefore fairly reliable.

Given an entity e and a phrase $d[s]$, we need to compute $p(e|d[s])$. Having no access to such a probability distribution, we choose to approximate this quantity by $\frac{p(e)}{p(d[s])}$, where $p(e)$ is the probability that e is linked to, and $p(d[s])$ is the probability that $d[s]$ occurs in a text. In other words, we estimate the popularity of the entity and the commonness of the phrase separately.

We estimate the popularity of an entity e by a log-linear combination of its number of statements n_e, site links s_e and its PageRank $\log p(e)$. The PageRank is computed on the entire Wikidata using statement values and qualifiers as edges.

The probability $p(d[s])$ is estimated by a simple word unigram language model that can be trained either on any large unannotated dataset[4].

The local compatibility is therefore represented by a vector of features $F(e, d[s])$ and the local compatibility is computed as follows, where λ is a weights vector:

$$\begin{aligned} F(e, d[s]) &= (-\log p(d[s]), \log p(e), n_e, s_e, 1) \\ p(e|d[s]) &\propto e^{F(e,d[s])\cdot\lambda} \end{aligned}$$

4.2 Semantic similarity

The issue with the features above is that they ignore the context in which a mention is found. To make it context-sensitive, we adapt the approach of [6] to our setup. The

[3] This is the distance between the start position of the later spot and the end position of the earlier one

[4] For the sake of respecting our constraint to use Wikidata only, we train this language model from Wikidata item labels.

general idea is to define a graph on the candidate entities, linking candidate entities which are semantically related, and then find a combination of candidate entities which have both high local compatibility and which are densely related in the graph.

For each pair of entities e, e' we define a similarity metric $S(e, e')$. Let $l(e)$ be the set of items that e links to in its statements. Consider a one-step random walks starting on e, with probability β to stay on e and probability $\frac{1-\beta}{|l(e)|}$ to reach one of the linked items. We define $S(e, e')$ as the probability that two such one-step random walks starting from e and e' end up on the same item. This can be computed explicitly as

$$\begin{aligned} S(e,e') = \beta^2 \delta_{e=e'} + \beta(1-\beta)(\frac{\delta_{e \in l(e')}}{|l(e')|} \\ + \frac{\delta_{e' \in l(e)}}{|l(e)|} + (1-\beta)^2 \frac{|l(e) \cap l(e')|}{|l(e)||l(e')|} \end{aligned}$$

We then build a weighted graph G_d whose vertices are pairs $(s \in d, e \in E[s])$. In other words, we add a vertex for each candidate entity at a given spot. We fix a maximum distance D for edges: vertices (s, e) and (s', e') can only be linked if $|s - s'| \leq D$ and $s \neq s'$. In this case, we define the weight of such an edge as $(\eta + s(e, e'))\frac{D-|s-s'|}{D}$, where η is a smoothing parameter. In other words, the edge weight is proportional to the smoothed similarity between the entities, discounted by the distance between the mentions.

The weighted graph G_d can be represented as an adjacency matrix. We transform it into a column-stochastic matrix M_d by normalizing its columns to sum to one.[5] This defines a Markov chain on the candidate entities, that we will use to propagate the local evidence.

4.3 Classifying entities in context

[6] first combine the local features into a local evidence score, and then spread this local evidence using the Markov chain (where $LC(d)$ is the vector of all $p(e|d[s])$ for each spot and candidate at that spot):

$$G(d) = (\alpha I + (1-\alpha) M_d)^k \cdot LC(d) \tag{1}$$

We propose a variant of this approach, where each individual local compatibility feature is propagated independently along the Markov chain.[6] Let F be the matrix of all local features for each candidate entity: $F = (F(e_1, d[s_1]), \ldots, F(e_n, d[s_n]))$. After k iterations in the Markov chain, this defines features $M_d^k F$. Rather than relying on these features for a fixed number of steps k, we record the features at each step, which defines the vector

$$(F, M_d \cdot F, M_d^2 \cdot F, \ldots, M_d^k \cdot F)$$

This alleviates the need for an α parameter while keeping the number of features small. We train a linear support vector classifier on these features and this defines the final

[5] The matrix M_d is a square matrix whose dimension is the number of candidate entities in the text. This can grow large but the matrix remains sparse due to the locality assumption.

[6] It is important for this purpose that features are initially scaled to the unit interval.

score of each candidate entity. For each spot, our system picks the highest-scoring candidate entity that the classifier predicts as a match, if any.

5 Experimental setup

Most entity linking datasets are annotated against DBpedia or YAGO. Wikidata contains items which do not have any corresponding Wikipedia article (in any language), so these items do not have any DBpedia or YAGO URI either.[7] Therefore, converting an entity linking dataset from DBpedia to Wikidata requires more effort than simply following `owl:sameAs` links: we also need to annotate mentions of Wikidata items which do not have a corresponding DBpedia URI.

We used the RSS-500 dataset of news excerpts annotated against DBpedia and encoded in NIF format [20]. We first translated all DBpedia URIs to Wikidata items[8]. Then, we used OpenRefine [7] to extract the entities marked not covered by DBpedia and matched them against Wikidata. After human review, this added 63 new links to the 524 converted from DBpedia (out of 476 out-of-KB entities).

We also annotated a new dataset from scratch. The ISTEX dataset consists of one thousand author affiliation strings extracted from research articles and exposed by the ISTEX text and data mining service[9]. In this dataset, only 64 of the 2,624 Wikidata mentions do not have a corresponding DBpedia URI.

We use the Wikidata JSON dump of 2018-02-24 for our experiments, indexed with Solr (Lucene). We restrict the index to humans, organizations and locations, by selecting only items whose type was a subclass of (P279) human (Q5), organization (Q43229) or geographical object (Q618123). Labels and aliases in all languages are added to a case-sensitive finite state transducer index.

We trained our classifier and its hyper-parameters by five-fold cross-validation on the training sets of the ISTEX and RSS datasets. We used GERBIL [20] to evaluate OpenTapioca against other approaches. We report the InKB micro and macro F1 scores on test sets, with GERBIL's weak annotation match method.[10]

Table 1 compares our approach to other entity linkers available online and compatible with the GERBIL evaluation platform. Our approach performs best on datasets of small text fragments (tweets in Microposts 2016, academic affiliations in ISTEX-1000) where context dependencies between entities are narrow. We suspect that the weakness of our language model (trained on Wikidata terms only) also hinders performance on longer texts.

6 Conclusion

The surface forms curated by Wikidata editors are sufficient to reach honourable recall, without the need to expand them with mentions extracted from Wikipedia. Our

[7] This is the case of Julie Pace (Q34666768) in Figure 1.

[8] We built the nifconverter tool to do this conversion for any NIF dataset.

[9] The original data is available under an Etalab license at `https://www.istex.fr/`

[10] The full details can be found at `http://w3id.org/gerbil/experiment?id=201904110006`

	AIDA-CoNLL		Microposts 2016	
	Micro	Macro	Micro	Macro
AIDA	**0.725**	**0.684**	0.056	0.729
Babelfy	0.481	0.421	0.031	0.526
DBP Spotlight	0.528	0.456	0.053	0.306
FREME NER	0.382	0.237	0.037	**0.790**
OpenTapioca	0.482	0.399	**0.087**	0.515
	ISTEX-1000		RSS-500	
	Micro	Macro	Micro	Macro
AIDA	0.531	0.494	**0.455**	**0.447**
Babelfy	0.461	0.447	0.314	0.304
DBP Spotlight	0.574	0.575	0.281	0.261
FREME NER	0.422	0.321	0.307	0.274
OpenTapioca	**0.870**	**0.858**	0.335	0.310

Table 1. F1 scores on test datasets

restriction to people, locations and organizations probably helps in this regard and we anticipate worse performance for broader domains. Our approach works best for scientific affiliations, where spelling is more canonical than in newswire. The availability of Twitter identifiers directly in Wikidata helps us to reach acceptable performance in this domain. The accuracy degrades on longer texts which require relying more on the ambiant topical context. In future work, we would like to explore the use of entity embeddings to improve our approach in this regard.

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: A nucleus for a web of open data. The semantic web pp. 722–735 (2007). https://doi.org/10.1007/978-3-540-76298-0$_5$2
2. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating Embeddings for Modeling Multi-relational Data. In: Advances in Neural Information Processing Systems. p. 9 (2013)
3. Cucerzan, S.: Large-scale named entity disambiguation based on Wikipedia data. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL) (2007). https://doi.org/10.1007/978-3-319-73706-5$_1$0
4. Ganea, O.E., Hofmann, T.: Deep Joint Entity Disambiguation with Local Neural Attention. In: Conference on Empirical Methods in Natural Language Processing (EMNLP) 2017 (Apr 2017). https://doi.org/10.18653/v1/D17-1277
5. Geiß, J., Spitz, A., Gertz, M.: NECKAr: A named entity classifier for Wikidata. In: International Conference of the German Society for Computational Linguistics and Language Technology. pp. 115–129. Springer (2017). https://doi.org/10.1007/978-3-319-73706-5$_1$0
6. Han, X., Sun, L., Zhao, J.: Collective entity linking in web text: A graph-based method. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 765–774. ACM (2011). https://doi.org/10.1145/2009916.2010019

7. Huynh, D., Morris, T., Mazzocchi, S., Sproat, I., Magdinier, M., Guidry, T., Castagnetto, J.M., Home, J., Johnson-Roberson, C., Moffat, W., Moyano, P., Leoni, D., Peilonghui, Alvarez, R., Vishal Talwar, Wiedemann, S., Verlic, M., Delpeuch, A., Shixiong Zhu, Pritchard, C., Sardesai, A., Thomas, G., Berthereau, D., Kohn, A.: OpenRefine (2019). https://doi.org/10.5281/zenodo.595996
8. Klang, M., Nugues, P.: Named Entity Disambiguation in a Question Answering System p. 3 (2014)
9. Kolitsas, N., Ganea, O.E., Hofmann, T.: End-to-End Neural Entity Linking. arXiv:1808.07699 [cs] (Aug 2018). https://doi.org/10.18653/v1/k18-1050
10. Mihalcea, R., Csomai, A.: Wikify!: Linking documents to encyclopedic knowledge. In: Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management. pp. 233–242. ACM (2007). https://doi.org/10.1145/1321440.1321475
11. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
12. Moro, A., Raganato, A., Navigli, R.: Entity Linking meets Word Sense Disambiguation: A Unified Approach. Transactions of the Association for Computational Linguistics **2**(0), 231–244 (May 2014)
13. Raiman, J., Raiman, O.: DeepType: Multilingual Entity Linking by Neural Type System Evolution. arXiv:1802.01021 [cs] (Feb 2018)
14. Ratinov, L., Roth, D., Downey, D., Anderson, M.: Local and Global Algorithms for Disambiguation to Wikipedia. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. pp. 1375–1384. Association for Computational Linguistics (2011)
15. Sorokin, D., Gurevych, I.: Mixing Context Granularities for Improved Entity Linking on Question Answering Data across Entity Categories. arXiv:1804.08460 [cs] (Apr 2018)
16. Spitkovsky, V.I., Chang, A.X.: A Cross-Lingual Dictionary for English Wikipedia Concepts. LREC pp. 3168–3175 (2012). https://doi.org/10.1007/978-3-642-10871-6$_7$
17. Štajner, T., Mladenić, D.: Entity resolution in texts using statistical learning and ontologies. In: Asian Semantic Web Conference. pp. 91–104. Springer (2009)
18. Strube, M., Ponzetto, S.P.: WikiRelate! Computing semantic relatedness using Wikipedia. In: AAAI. vol. 6, pp. 1419–1424 (2006)
19. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: A core of semantic knowledge. In: Proceedings of the 16th International Conference on World Wide Web. pp. 697–706. ACM (2007)
20. Usbeck, R., Eickmann, B., Ferragina, P., Lemke, C., Moro, A., Navigli, R., Piccinno, F., Rizzo, G., Sack, H., Speck, R., Troncy, R., Röder, M., Waitelonis, J., Wesemann, L., Ngonga Ngomo, A.C., Baron, C., Both, A., Brümmer, M., Ceccarelli, D., Cornolti, M., Cherix, D.: GERBIL: General Entity Annotator Benchmarking Framework. In: Proceedings of the 24th International Conference on World Wide Web - WWW '15. pp. 1133–1143. ACM Press, Florence, Italy (2015). https://doi.org/10.1145/2736277.2741626
21. Vrandečić, D., Krötzsch, M.: Wikidata: A free collaborative knowledge base. Communications of the ACM (2014). https://doi.org/10.1145/2629489
22. Weichselbraun, A., Kuntschik, P., Braşoveanu, A.M.: Mining and Leveraging Background Knowledge for Improving Named Entity Linking. In: Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics. pp. 27:1–27:11. WIMS '18, ACM, New York, NY, USA (2018). https://doi.org/10.1145/3227609.3227670
23. Witten, I.H., Milne, D.N.: An effective, low-cost measure of semantic relatedness obtained from Wikipedia links (2008)
24. Xiao, H., Huang, M., Zhu, X.: TransG : A Generative Model for Knowledge Graph Embedding. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 2316–2325. Association for Computational Linguistics, Berlin, Germany (2016). https://doi.org/10.18653/v1/P16-1219