Evaluating Web Table Annotation Methods: From Entity Lookups to Entity Embeddings (ISWC 2017 Paper) - Supplementary Material

Data Citation (or feel free to cite the ISWC paper):

- *Efthymiou, Vasilis; Hassanzadeh, Oktie; Rodríguez-Muro, Mariano; Christophides, Vassilis (2017):* ***Evaluating Web Table Annotation Methods: From Entity Lookups to Entity Embeddings****. figshare. https://doi.org/10.6084/m9.figshare.5229847*

This document describes implementation details of our paper entitled "Matching Web Tables with Knowledge Base Entities: From Entity Lookups to Entity Embeddings".
This is our best effort to describe implementation details and make our results reproducible while we work on making portions of our code base open-source or available via limited APIs for public use. Please feel free to contact us if you have any questions.

Datasets can be found at: http://www.cs.toronto.edu/~oktie/webtables/ or https://doi.org/10.6084/m9.figshare.5229847
Refer to the README file for details: http://www.cs.toronto.edu/~oktie/webtables/README.txt and please feel free to contact the authors if you have any issues with the data sets.
NEW: The embeddings model binary file can also be found at http://www.cs.toronto.edu/~oktie/webtables/embeddings-model/

FactBase Lookup Details

Summary of implementation details

- Our FactBase index is actually an integration of Wikidata, DBpedia, Freebase, and YAGO, solely based on the Wikipedia URLs, and only keeping those URIs that are associated with an English Wikipedia page. We have a "FactBase Unified" index which is only the basic facts (label, description, types) that are derived from Wikidata, and a "FactBase Full Facts" index that contains all the facts from all the ingested knowledge bases. Examples of the JSON objects stored in these indices are shown in Appendix 1.

- For the experiments in our paper, we used the "FactBase Unified" index (i.e., Wikidata basic facts and types). Wikidata version used in the API used in the experiments is:
  - Downloaded on: April 11, 2016
  - Last modified date on server: February 1, 2016
  - Download link: http://tools.wmflabs.org/wikidata-exports/rdf/exports/20160201/

- For the type lookups performed in the refined FactBase lookup method described in the paper, we get all the types in the unified index (i.e., Wikidata types) with the exception of "Wikimedia disambiguation page" & "Wikimedia category".

- We keep the top-5 most frequent types, and only look at the direct types, i.e., we do NOT use subclass hierarchy in our implementation. We considered doing so but as you can see in the Wikidata examples below, the direct types were useful as-is. Our algorithm can be modified to use subclass hierarchy if needed, although there needs to be a proper stop criteria to avoid identifying top-level classes such as owl:Thing as the common type.

- Regarding the number of agreements for the identified relation in FactBase lookup, the number that we found to work well in all cases was 5, i.e., when 5 verified lookup results contain the same relation between the entity described in a row and an entity mentioned in a specific column of the same row, we consider that this relation expresses the semantic relation between all the entities described in the rows of this table and the entities described in this specific column.

Backend Details

- Our index uses Riak https://en.wikipedia.org/wiki/Riak which is a key-value store with embedded search. We have various key-value indices in Riak, e.g. to lookup facts given a URI (NTriple statements that their subject is the URI), and JSON objects to facilitate search. Appendix 1 shows examples of such JSON objects.

- Riak comes with embedded Solr search http://lucene.apache.org/solr/ so we can query for labels and other facts, as needed for our refined FactBase lookup implementation.

- The Riak index is populated in parallel (in MapReduce, using BigInsights) using RDF data dumps (e.g., Wikidata dump mentioned above)

Embeddings Method Details

Summary of method. The core algorithm is described in Sections 4 and 5 in [Zwicklbauer et al., 2016]. Note, we only use the RDF-KB method for entity embeddings (for the reasons described in the answer to Reviewer #4).

The embeddings model binary file can also be found at
http://www.cs.toronto.edu/~oktie/webtables/embeddings-model/

Summary of corpus and embeddings generation details (off-line):

- Alpha = 0.1 (probability of random jump through the graph)

- Model used: **skip-gram**, 300 dimensions, default word2vec parameters (original implementation found here: https://github.com/dav/word2vec) (Skip-gram: the input to the model is a word wi and Word2Vec predicts the surrounding context words wi−2,wi−1,wi+1,wi+ 2.)

- Similarity computation: cosine

Summary of candidate index generation (off-line):

- Candidate generation from rdf:label(s) and lexvo:label (from DBpedias Lexicalizations dataset http://oldwiki.dbpedia.org/Datasets/NLP) :
- Other processing: stemming, stopword removal, trigram similarity > 0.82

Summary of disambiguation implementation details (on-line):

- Random walk strategy: PageRank
- Edge weights are similarities (cosine similarity) of the embeddings for the two entities (normalized to [-1,1]) and represents likelihood to walk from a node to the adjacent
- Random walk (50 PageRank iterations) gives node relevance: average amount of visits.
- If 2 nodes have same label, the most commonly referenced node is preferred (but this is just one factor, similarity is the most important one)

As per a question from our paper's reviewers, there are alternative models can be used that are based only Wikipedia text:

- https://github.com/idio/wiki2vec/
- https://github.com/3Top/word2vec-api#where-to-get-a-pretrained-models

Note that our model differs from the above in that it does not rely on a rich textual source like Wikipedia and so can be generalized by using other RDF & Linked Data sources and even enterprise databases.

Blocking for Ontology Matching Based Method Details

Summary of method. The blocking method used for Ontology Matching is Token Blocking [Papadakis et al., 2011] applied to all DBpedia entities: Each word (aka token) in a DBpedia entity description defines a block (an index position), and each description having this token is placed in this block. Those blocks are stored in a distributed NoSQL key-value database (http://basho.com/products/riak-kv/), in which keys are tokens and values are lists of all the DBpedia URIs having a specific token.

After creating the index, we can search for all the entries (DBpedia URIs) that correspond to a specific key (token). We apply this search service for all the tokens in the label column of a Web table to retrieve candidate annotation results.

We are currently considering more advanced blocking methods, such as the ones described in [Christophides et al., 2015].

References

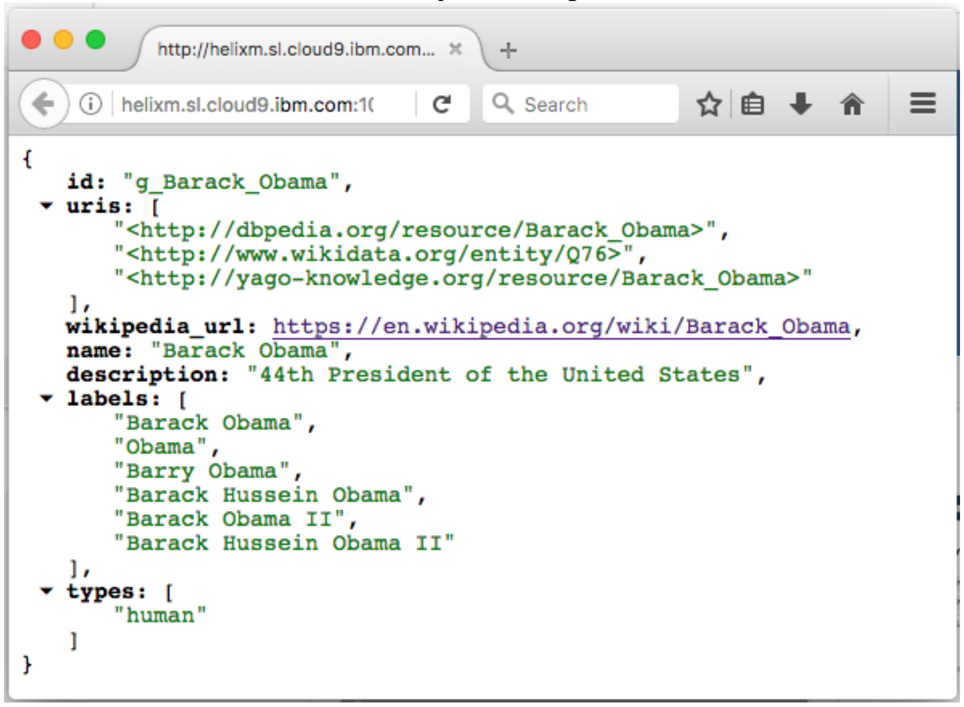
Stefan Zwicklbauer, Christin Seifert, Michael Granitzer: DoSeR - A Knowledge-Base-Agnostic Framework for Entity Disambiguation Using Semantic Embeddings. ESWC 2016: 182-198 [link]

George Papadakis, Ekaterini Ioannou, Claudia Niederée, Peter Fankhauser: Efficient entity resolution for large heterogeneous information spaces. WSDM 2011: 535-544 [link]

Vassilis Christophides, Vasilis Efthymiou, Kostas Stefanidis: Entity Resolution in the Web of Data. Synthesis Lectures on the Semantic Web: Theory and Technology, Morgan & Claypool Publishers 2015 [link]

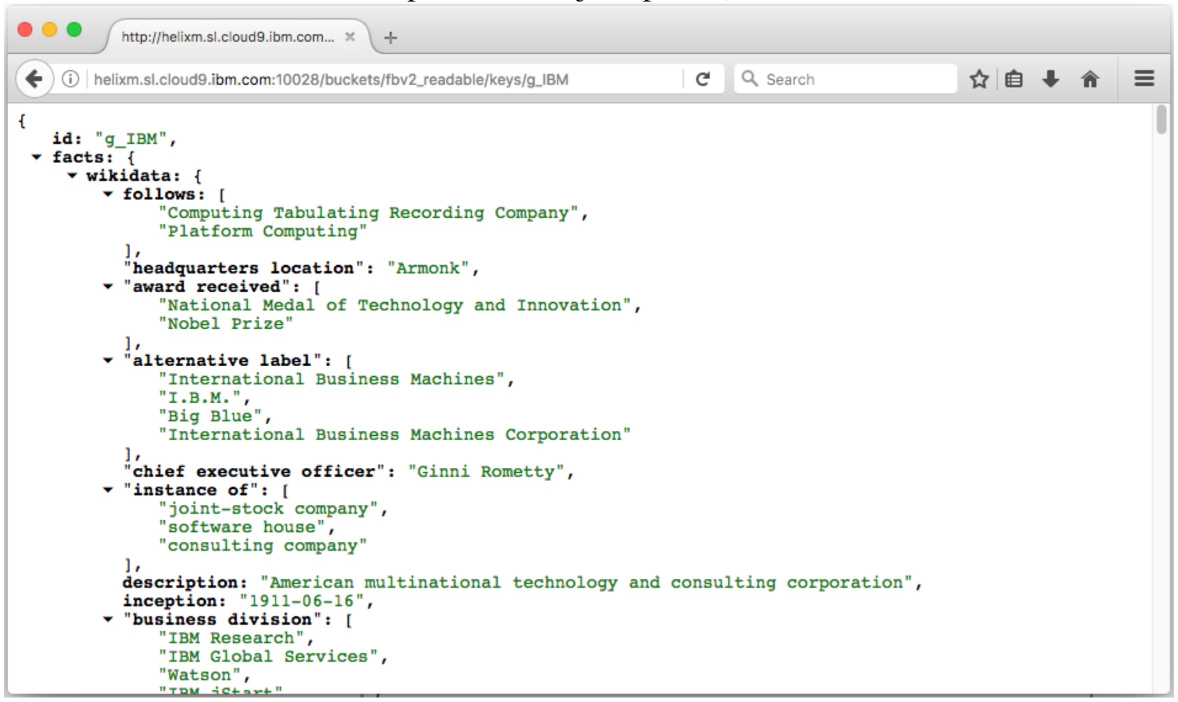
Appendix 1 - FactBase API Examples

FactBase Unified Index JSON Object Example for "Barack Obama"


http://helixm.sl.cloud9.ibm.com...
helixm.sl.cloud9.ibm.com:1(
Search

```
{
    id: "g_Barack_Obama",
  ▾ uris: [
        "<http://dbpedia.org/resource/Barack_Obama>",
        "<http://www.wikidata.org/entity/Q76>",
        "<http://yago-knowledge.org/resource/Barack_Obama>"
    ],
    wikipedia_url: https://en.wikipedia.org/wiki/Barack_Obama,
    name: "Barack Obama",
    description: "44th President of the United States",
  ▾ labels: [
        "Barack Obama",
        "Obama",
        "Barry Obama",
        "Barack Hussein Obama",
        "Barack Obama II",
        "Barack Hussein Obama II"
    ],
  ▾ types: [
        "human"
    ]
}
```

FactBase Full Fact Index Example JSON Object (partial)


http://helixm.sl.cloud9.ibm.com...
helixm.sl.cloud9.ibm.com:10028/buckets/fbv2_readable/keys/g_IBM
Search

```
{
    id: "g_IBM",
  ▾ facts: {
      ▾ wikidata: {
          ▾ follows: [
                "Computing Tabulating Recording Company",
                "Platform Computing"
            ],
            "headquarters location": "Armonk",
          ▾ "award received": [
                "National Medal of Technology and Innovation",
                "Nobel Prize"
            ],
          ▾ "alternative label": [
                "International Business Machines",
                "I.B.M.",
                "Big Blue",
                "International Business Machines Corporation"
            ],
            "chief executive officer": "Ginni Rometty",
          ▾ "instance of": [
                "joint-stock company",
                "software house",
                "consulting company"
            ],
            description: "American multinational technology and consulting corporation",
            inception: "1911-06-16",
          ▾ "business division": [
                "IBM Research",
                "IBM Global Services",
                "Watson",
                "IBM iStart"
```