WASEDA University

SHEN, Shuyang <shua@akane.waseda.jp>

Future Research

Mizuho Iwaihara <iwaihara@waseda.jp> 2022年10月26日 14:02
To: "SHEN, Shuyang" <shua@akane.waseda.jp>

Hi,

The file of WikiTable datasets contains target column only? How about other columns, and column headers? Also, the DoDuo paper says they do multi-tasking on WikiTable that includes column relationships. Are there column relationship data? Could send me more samples and metadata of WikiTables?

For the file you send, the followings are observed.
- There are many tuples that have only general types, such as person.person, location.location, organization.organization. In fact, many tuples are person.person. This suggests type labeling of WikiTable is coarse-grained. The paper says typing of WikiTable is automatically done by some rules, which may be lacking preciseness.

So here is an interesting research topic, of giving finegrained types on WikiTable dataset.
Here is how to find fine-grained types.
1. Using entity names in the column, and column/table headers, identify their Wikipedia categories. Wikipedia API, and Wikipedia dump (4GB) can be used to locate the original table that holds the entity names. From the source of the table page, we can find its category labels.
2. Caligraph can be used to lookup the category of the table. Caligraph also gives the DBpedia types of the categories.

We expect the DBPedia types above are more fine-grained than WikiTable.
We know that Caligraph contains 15% errors, and Zhang Zhenyang is constructing a better DBpedia type mapping, but for now Caligraph is sufficient to identify the DBpedia types of the table column.
We aggregating types of entities into the column types, which requires a sophisticated algorithm. Majority voting is the simplest one (not best).

3. We evaluate accuracy of the column typing result, by our own manually labeled column types. Here, Wikitable cannot be used as ground-truth, because we assume WikiTable is not finegrained.

4. For evaluation metrics, checking F1-score on exact match on the DBPedia type is used in the literature. But for finegrained types, exact hit becomes harder. Here we can use a relaxed way of evaluation such that scoring the typing results by the number of hops (edges) from the golden DBPedia type (or with some weighting). Then we can give points to near-hits.

Technical challenges are to correctly identify set of DBpedia types for the target column, from member entities of the column, their wikipeida articles, and Caligraph. We need to combine search on category graphs and semantic features that BERT can give. For identifying finegrained types, language model-based approach like DoDuo/BERT has limitation. This is more like an information retrieval problem on category hierarchies, where a dense retriever used in question answering is more suitable.

Numerical attributes can be typed using BERT, like DoDuo.. But its more detailed attribute type needs to be discovered from Caligraph.

5. An alternative task is, assuming that the column typing result by the above using Caligraph is correct, and use this as the ground-truth. Then we predict column types of an unseen table. The unseen table can actually be an existing Wikipedia table that is held out from the training set. Then let the model predict the column types. This is a type of distant supervision, and we can skip costly manual labeling. This is a scenario of column typing for a new table in Wikipedia that lacks proper category types. The new tables may contain new unseen entities as well as entities already existing in Wikipedia. We can compare against DoDuo.

I think the above task is well enough for you master's work, and good to start.
[元のメッセージ非表示]

--
岩井原　瑞穂（IWAIHARA Mizuho）iwaihara@waseda.jp

早稲田大学　大学院情報生産システム研究科
〒808-0135 福岡県北九州市若松区ひびきの2-7
TEL&FAX： 093-692-5217