Progress Report

Vinh Thinh Ho

December 16, 2019

Below, I would like to give a short summary about my current research project Quantity Search. First, I give a report about my current progress working on this project, and second, I define some goals to achieve in the next year.

1 Current Progress

The objective of the project is to answer quantity-related queries such as: *"cars with price less than $100k"*. There are also more complex queries that we want to answer, however we could leave that for future work.

In this year, we got a full paper plus a demo paper on quantity search over text, and now we want to extend our work to a new data source: Web Tables, as we see that the amount of quantities on web tables is very huge. In this work, I will call our system "TabQs"

The detail of our current progress for TabQs has been described in the write up. Now I will briefly describe the main idea of the work, as well as list some problems that we are facing now.

1.1 Main Idea

Given a web table, to understand it completely, we may also need its surrounding context. Table context might be: caption, page title, page content. Different parts of the context could be weighted differently.

Now to extract Qfacts from the table, we do the following steps:

- *Quantity recognition*: There are many options and tools here. One could develop her own rule-based parser, or use an external tool. As we are working on a IR problem, recall is important, and building our own rule-based parser is totally not enough. Hence, we use a combination of two state of the art tools for detecting quantity: QEWT + Illinois Quantifier (as described in the write up).

- *Entity recognition*: Previous works have been proposed for entity disambiguation in web tables, however reusing them is very difficult, I tried to reuse the code, but totally difficult, and we have to reimplement them. Anyway, previous works rely on co-occurence of textual cues of the tables

on external sources (i.e. text, KB), and we hypothesize that in case we focus on quantitative tables, with very few texts, these techniques do not work well. So, in this entity recognition step, we only find candidate entities for each entity mention, using a prior model, built from hyperlinks from various sources. The disambiguation will be done jointly later on.

- *Column linking*: We find the relation between quantity and entity columns, and along with this we do disambiguation of the entities. We call this the "Table Plausibility Maximization" problem. Our object function has two main sub-functions:
  - **Homogeneity**: describes that entities of the same column should have same types. We model the type of a entity column as a bag of types with frequency. So that we compute the homogeneity as the *entropy* of this bag.
  - **Connectivity**: describes the likelyhood that the quantity column connects to the entity column (using current entity assignment). And the likehood score is pre-trained from a neural-based model (as described in the write up).
- *Find context:* After the connection between entity and quantity columns are determined, we need to find their context. Of course, we know that the header of quantity column is vital (without it we cannot even understand the quantity column, as they are pure quantities). However what about other tokens in the table's context? Let's look at following ways:
  - *Map entity-quantity from table to text*: As I demonstrated a long time back, one strategy is to match entities and quantities from table to text, to find their context co-occured, however I observed that this is very diffcult (matching quantities must be approximated) and hence noisy, and not scalable (slow in performance and inefficient in space).
  - *Using entire context.* Table context is divided into different parts: caption, page title, page content, and we have different strategies of choosing context that include them or not. This is not systematic, however might work well, hence we might end up using this way?
  - *Search on the fly.* If we start from the query, do searching on the fly, we can somehow take into account the surrounding context of the table without actually need to extract the full Qfacts's context and offline indexing like we did for Qsearch. This way is good because we can filter out bad tables by considering their surrounding context first (here we can even apply boosting to weight the importance of each context part, e.g., title and caption are more important than page content), then we do Qfacts extraction on the table itself.
- *Search*: We will basically use the ranking models from Qsearch, plus some modifications to fit with our settings.

2 Future Plans

2.1 Technical Plans

So far I have summarized what I have done. There are still many things to do here.

- First, I think I am not so good for pure theoretical research – building an awesome methodology first, but then fail on evaluation. I think the data is crucial, so what we think might not be what the data thinks, and hence I like more when having a bad running system first, then improve over it (with theories and methodologies). So in my opinion, the first step to do is to complete the implementations that I have been doing so far. I feel motivated when seeing something running, even it is bad.
- Second, parallel with the first step is to define the concrete evaluations that we will do, along with baselines that we will compare to. There are 3 tasks here:
  - *Entity disambiguation*: as you mentioned, this could be not our main focus, but we can show a few numbers here.
  - *Column linking:* we need to start by collecting datasets. I will have a look at other papers to find an appropriate datasets if available, otherwise we will collect our own test set. The former option should be preferred.
  - *End-to-end search*: most important, and most difficult to conduct experiments. Questions:
    – what do we show to evaluators? entities (group/not-group by tables), tables, table context?;
    – how to we show to evaluators? (because we have a huge amount of information to show for each entity result here) – which platforms?;
    – test queries? can we reuse the test queries from Qsearch, or do we have to collect more? This very relies on the data we have.

2.2 Non-Technical Plans

I would like to submit something next year. I am not sure if this plan is possible, since there are still many things to do. The earliest deadline we could thing about is SIGIR as suggested by you, which is in roughly 1 month, and the next one is CIKM in May. So ideally, we could submit to both of the conferences without conflict. I will try with SIGIR first. Another option could be VLDB, however SIGIR is probably a better choice.

If we can submit a paper on quantity on web tables, the next research project could be either:

- *Handle complex queries*: We extend our work with more complex queries, e.g., queries with more than 1 quantity condition

- *Quantity consolidation*: The same entity and context can have different quantites reported from different source, so we could do clustering / aggregating to get the best quantity answer.
- *Quantity linking*: Reasoning quantities across sentences in text. This direction is best suit for the finance domain. E.g., Given sentence *"BMW i3 has price of $100k in 2019. The figure for BMW i8 is $20k more expensive"*, we want to link $20k with $100k, and resolve the value to 120k with the context "price".
- *Quantity search with non-KB results*: Current works focus only on entities in KB. I think that working with non-KB entities are also very important, especially when we want to search for products with different models or versions.