Next-Gen Search Lab Report

Mingwei Huang (mingwei6@illinois.edu)

August 25, 2021

1 Problem

The purpose of the research is to build a new form of academic search engine. Almost everyone has encountered the scenario that one has little knowledge about the problem that is trying to be solved, and not all users are familiar with the field that they are searching so that there can be many keywords they are looking for but they have never heard about. And it takes a lot of time to get a glance of the full image of the problem. Furthermore, during such process, there is a chance that users can never find some information that can be useful to them since they have not thought about the problem in that perspective. Thus, the new academic search engine aims to help users grab the information about the problem they are tackling even though they do not have much idea about it.

For example, if I am working on some indexing problem and trying to order the data in a way that it is always efficient to retrieve part of them satisfying certain conditions, e.g. the sum is equal to some number. But if I do not know much about indexing, then it can be very hard to do the research since I do not have a good clue on how to start it. If I search with the query "data order", it will be very unlikely for me to know this problem is a partial sum problem. Current academic search engine, like Google Scholar, neither interpret any specific problem nor give a clue of the possible solutions if any. The most feasible thing is to do some search on Google and it may return some pages of Stack Overflow or some others that contains similar specific problems. However, the current approach relies on volunteer answering and still not covers many problems, which is very inefficient use of information. With the development of NLP, it starts to become possible to tackle this problem.

The form of the search engine is to read some problem description as the input and return pieces of information that contains some possible solution or hints of them. I choose to restrict the problem description to be as general as several words since the dataset we want to use, academical papers or tech articles/posts, do not contain a problem-solution structure, and even though some of them do, the problems tend to be very general. It is too difficult for current NLP to generalize specific problem descriptions and match them to papers mentioning abstract problem-solution pairs. After the decision of using general problem description, I then need to deal with the the problem immediately following it: there may be hundreds of potential solutions that match the input problem, thus, there must be some navigation to help the users locate their wanted solutions.

Therefore, I come into the general design of the form of the search engine which contains two major part: the first part is on-line, taking an input query and returning some concepts/keywords that are highly similar/relevant to the query, and the second part is off-line, it is a navigable map where users can navigate the map from the returned concept and look for the solutions/sub-problems/similar problems and finally locate the paper/article they need.

2 Data

Metadata(including abstracts) of all ArXiv papers.

3 Approaches

The implementation of the search engine can be separated into two parts: query projection, projecting a general problem description into the concept space where the coordinate is the degree of relatedness of each computer science concept, and a concept map, which is a knowledge graph containing all computer science concept entities and their relations.

3.1 Query Projection

My approach here focus on the occurrences of the query words in the data, and tries to extract related concept entities based on their co-occurrences. A query parser is not implemented but definitely helpful in this case.

Denote the possibility of a latent relation $z \in Z$ between a query word $w \in V$ and a concept entity $t \in T$ as $p(z|w, e)$.
I set the following naive assumptions:

1. The presence of a word depends only on an existing concept:
   for all $w \in V, p(w) = \int_t p(w|t)dt$.
2. The probability of the co-occurrences depend only on the relation z: for all $t_1, t_2 \in T$, $\mathcal{D}(p(w|t_1, z)) = \mathcal{D}(p(w|t_2, z))$ where $\mathcal{D}$ stands for the probability distribution.

Hence, we can then denote $p(w|t, z) := p(x|z)$ and $p(z|x) = p(z, t|w)$.

Since it is almost impossible to precisely filter out meaningful relation z with only information of $p(x|z)$, I choose to select z in the area with high density of useful relations. Intuitively, if $p(x|z)$ is high, since $p(x) = \mathbb{E}_{z \sim p(z|x)}[p(x|z)]$ and $\mathcal{D}(p(x|z))$ is constant for z by the assumption, we have $p(x) \sim p(z|x)$ so $p(z|x)$ is high, and z is very likely to be meaningful.

Lastly, because $p(x) = p(w|t) \propto \frac{\text{co}_D(w,t)}{\text{freq}_D(t)}$ where co and freq are co-occurrences and frequencies respectively, I come up the following ranking function:

$$\text{Score}_q(t) = (1 - \alpha)P_q(t|D) + \alpha P(t|\bar{S}),$$

where

$$P_q(t|D) = \sum_{d \in D} P_q(t|d),$$

$$P(t|d) = \frac{\log \text{bm25}_d(q) \cdot \text{freq}_d(t)}{\text{freq}_D(t)^{k1}},$$

and

$$P(t|\bar{S}) \equiv \| \text{diff}(t, S) \|_a^b + c = c + \sqrt[b]{\sum_{s \in S} \text{sim}(t, s)^{-a}}.$$

Here, $P(t|\bar{S})$ measures the difference between the valuing the concept entity t and the existing outputs so that the result outputs can be more diversified. The similarity between two words is calculated by comparing their word embedding which is also trained from the data set.

3.2 Concept Map

The concept map is treated as a knowledge graph where the nodes are only computer science keywords and the edges are relations. The result that I am expecting to be should have similar form with the knowledge graph in the SciIE dataset, and an example is shown on the right. There should be only a few types of entity and relations and it should be also simple to find evidence(specific papers) for such relations. However, different from the data in SciIE, the ArXiv papers in general have more complex structure and needs more abstract logical reasoning to accurately extract the correct relations. The performance of current NLP IE model on such task is not yet to be examined.

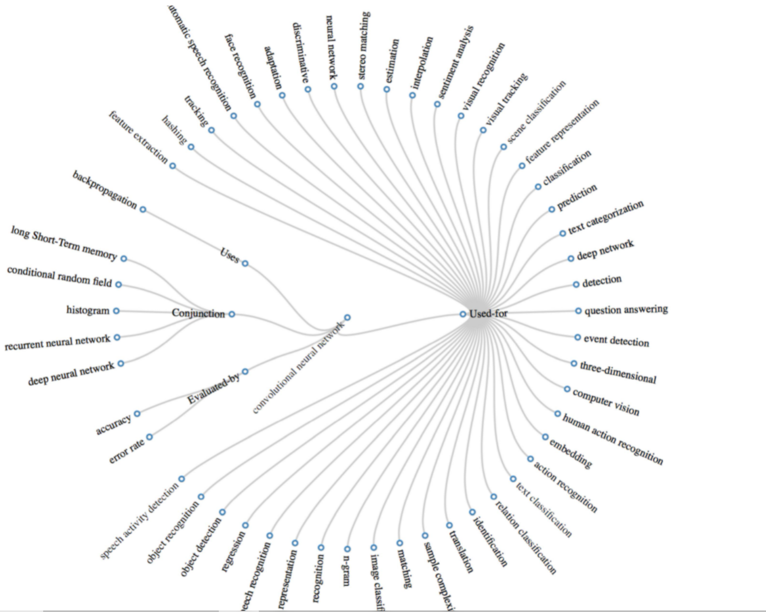

Figure 1: Part of the KG connecting to "CNN"

3.3 Supplement

It can also be helpful to further explain relations between concepts and problems as well as relations among concepts. A simple way is to find surrounding words of a concept that are likely to appear when the query is also present. Those words can give a hint about how the concept word is used and its context if there is sufficient data containing both of them. For instance, if we search "medicine" and get a concept entity "data", we can look into the contextual words around "data" to get more information about how "data" is related, and the result contains words like "patient", "cancer", and "medical" which can well explain its use.

Another direct implementation is to generate a sentence that concisely but precisely reveals those relations. There is research going on for this task, but it requires definitive sentences as text input and an existing knowledge graph to help the language model reasoning. The definitive sentences can be extracted by doing text sentiment analysis from text in tech articles and Wiki, but the latter one can only be done after completing a concept map.

4 Assessment

I manually test about thirty queries for the query projection and often 8 of top 10 concepts make complete sense to me. I do not have a dataset, e.g. a computer science keyword knowledge graph, which can be used to accurately measure the precision and recall for the ranking function, but I think build an alpha-version search engine can gather some data to evaluate the accuracy and improve the model. I have a simple template code for the website which can accessed at the end of the paper.

Other functionalities are still in the researching stage with no available implementation yet.

5 Reflection

I think the research experience is quite challenging and fun. The research task does not have a bound and I have learned a lot about NLP during the research. I don't know anything about NLP before the research, but now, I'm able to read all the current papers in NLP field, and I get a general glance of most NLP tasks and their current solutions.

Maybe some existing pre-trained models can be directly used for the research project, and it will be helpful to give some recommendations and provide the access to the machine to run them.

6 Future Plan

If we continue developing this search engine, it is necessary to build a concept map and it requires a complicated language model to process the texts. And it will be difficult but interesting to use a model to complete the task, by either thinking about the approaches to extract relations involving logic reasoning or constructing some dataset for the model to be trained on.

7 Code Archive

For more information, refer to https://github.com/Scott-Huang/Academic_Engine.