AI systems for query understanding and response

Rohan D. Kekatpure

1. Problem description

We consider the general problem of building an AI agent capable of understanding human queries and providing resources (URLs) as responses. One could consider this problem to be solved. After all, isn't that precisely what search systems like Google and Bing do? Yes, and in that sense this writeup could be considered an analysis of those systems as opposed to a novel design.

If you Google the phrase "how to throw away a bad habit", none of the first 10 results contain the words "throw away". The titles of the returned URLs seem more relevant than the phrasing of the original question. How does Google extract deeper meaning from my poorly phrased question? In this writeup we consider possible ways in which Google could be doing this.

2. What makes the problem difficult

Two main aspects make the problem difficult. First is extracting semantic meaning from ambiguities present in human language. This was covered well in the lecture on *Understanding* and does not bear repeating here. Second is the problem scale: the web contains an estimated 50 billion indexed URLs, each of which is a candidate solution. Even a naive approach of returning URLs with the closest word-similarity would return *something*. The problem is it wont be relevant.

To provide relevant results, the phrases have to be mapped to more encompassing language elements ("throw away" → "get rid of" vs "propel"). But, as seen in the lecture on *Understanding*, this mapping is highly context dependent. Also, manual generation of this mapping is a non-starter. The only scalable way to build this mapping is from the queries and results themselves.

3. *Understanding* to the rescue

The lecture on *Understanding* provides a starting point. Lets assume that we choose to build this system using frames-based knowledge representation. Further assume that initially the frames are either blank or sparsely populated. Our task is to generate more frames from the queries and populate the existing frames using the responses that are chosen.

Assuming we have a good semantic mapping of words to their meta-categories for every query context, then the task is to simply retrieve the documents matching the frame, sort them by a suitable score (such as TFIDF) and return the results.

However, we *don't* have a good mapping initially (e.g. Google in the early days) and it must be built. The central concept that will help us build this set is that of "enrichment through feedback".

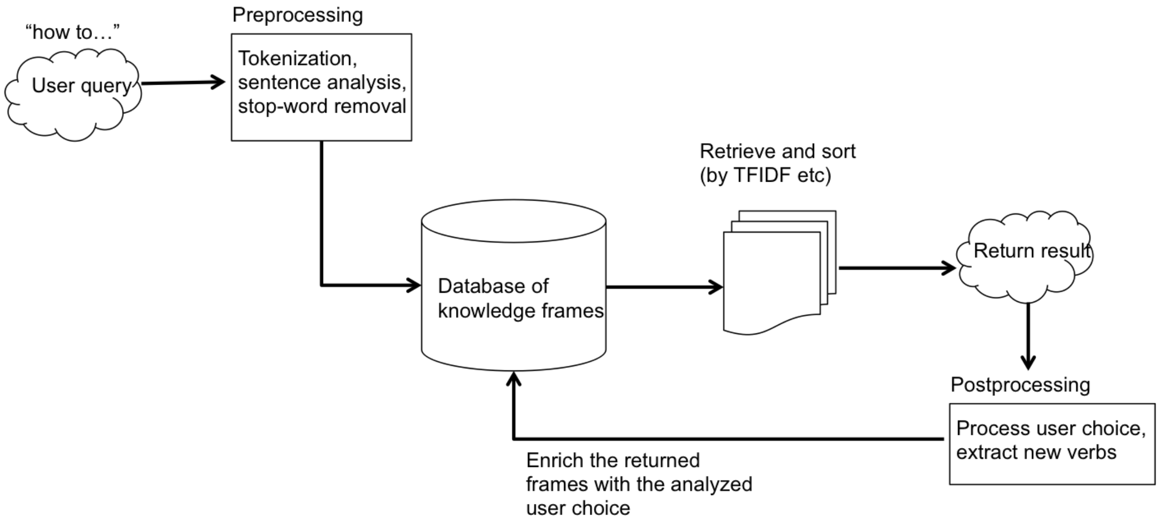
"how to…"
User query
Preprocessing
Tokenization, sentence analysis, stop-word removal
Retrieve and sort (by TFIDF etc)
Database of knowledge frames
Return result
Postprocessing
Process user choice, extract new verbs
Enrich the returned frames with the analyzed user choice

Fig. 1. Architecture of a query response and knowledge enrichment system.

4. Enriching knowledge frames through feedback

Figure 1 provides a conceptual architecture of a query understanding and response system that enriches its knowledge-base via analysis of user feedback. Human users interact with the system by making a query and consuming the results. The system, in turn, provides results by searching its existing knowledge base and incorporating the analyzed results of user choice.

By allowing such a system to run for a reasonable amount of time, one can expect the frames to stabilize their representation of a concept. Continuing our example query of "throwing away a bad habit", let us assume that the pre-processing step tokenized the query, removed the stop words, and created a question frame (serialized as a JSON) as follows:

```
{
  "agent": null
  "context": "habit"
  "verb": "throw"
  "meta-verbs": {"propel": 0.95, "other": 0.05 }
}
```

Initially, the system is has only one dominant mapping of the verb 'throw' to 'propel'. Consequently, most results are returned based on 'habit' and aren't relevant to the query. However, the user selects few webpages that seem slightly relevant to getting rid of bad habits. The system parses the test and extracts other verbs from these retrieved webpages and uses this data to enrich the frame. For example, the words in the user-selected webpages contains other phrases like 'kicking bad habits', 'get rid of smoking' and so on. The system computes new weights for the meta-verbs and inserts them back into the database. The enriched frame for the verb 'throw' might look like:

```
{
  "agent": null
  "context": "habit"
  "verb": "throw"
  "meta-verbs":
    {"propel": 0.65, "discard": 0.3, "kick": 0.05 }
}
```

While the dominant mapping is still incorrect, the weights for relevant verbs are increasing. It is not hard to imagine the weights converging toward relevant meta-verbs after a few hundred thousand queries.

The frame-enrichment task is most likely a “batch” task. I.e. the new frames aren’t immediately available for retrieval; they must be enriched in a nightly analysis run in a distributed computing cluster and available the next morning.

5. Summary

We considered conceptual design of a query response and knowledge enrichment system using *Understanding*. We chose frames as our knowledge representation and outlined how those frames can be populated with useful content at scale. The design presented is at a very high-level and glides over may practicalities such including sentence analysis, element (noun, verb) extraction, stopword removal, HTML parsing and distributed document retrieval. These are, however, solved problems with well-known open-source solutions available. Our focus here was on knowledge-enrichment which is essential to use *Understanding* for building practical systems.