Supplementary Material

provided for the article
"Olelo: a web application for intuitive exploration of biomedical literature"
Nucleic Acids Research Web Server Issue 2017

Milena Kraus , Julian Niedermeier , Marcel Jankrift , Sören Tietböhl ,
Toni Stachewicz , Hendrik Folkerts , Matthias Uflacker , Mariana Neves
Hasso Plattner Institute, August-Bebel-Str. 88, Potsdam, 14482, Germany

The supplementary material provides a more detailed description of the methods behind the Olelo question answering (QA) system. All components and resources are integrated and implemented into an in-memory database. We start by describing the database, followed by the biomedical resources that we integrated and the NLP components of the QA system. A diagram of our system is illustrated in Figure 1.

1 In-Memory Database

Conventional databases store the data on hard disks, which must be accessed with every query to the system. In order to increase efficiency we rely on an IMDB, which stores the data in main memory and keeps the input and output operations away from the hard drives [1]. Further advantages of the IMDB technology include: multi-core processing and parallelization, column-based data layout, lightweight compression and partitioning. These features support storage of large indexing tables (cf. below), while parallelization can be used to accelerate QA procedures within the database. We rely on the SAP HANA database with a total main memory size of 2048 GB. We integrated the domain resources into the database and implemented all our QA components as SQL procedures.

2 Integrated Resources

Our system relies on four resources for biomedicine: the MeSH ontology (http://www.nlm.nih.gov/mesh/), terminologies from the UMLS database, PuMed abstracts and PubMed Central Open Access (PMC OA) full texts. The MeSH ontology and UMLS terminologies are used for compiling dictionaries for the named-entity recognition (NER) approach (cf. below) as well as for navigation purposes in the Web application.

3 Question Answering Components

The QA process starts by understanding the input question, followed by extracting relevant documents and passages and finally returning an appropriate answer, e.g., an exact answer or a summary. In this section, we describe each of these components of our QA system in detail.

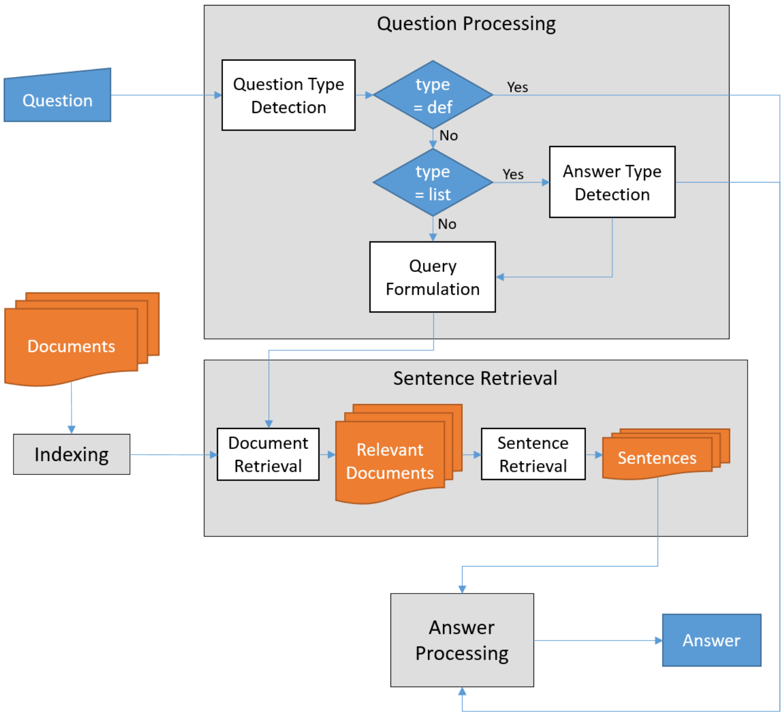

Figure 1: Work-flow of our question answering system.

3.1 Indexing

To get results quickly when looking into a large document collection, these must be previously indexed. A full-text index can provide important information from unstructured texts, and are indispensable for document and sentence retrieval. We indexed titles and abstracts from PubMed as well as the full text of PubMed Central Open Access documents, all of which can be accessed in our QA system. For each of these collections, we built a NER and a linguistic full-text index. The NER index was created based on dictionary-based matching provided by the IMDB. This function is powered by custom dictionaries that we created by compiling terms, names, synonyms and variants from both MeSH ontology and UMLS terminologies. On the other hand, the linguistic index performs some basic linguistic processing on the text, such as tokenization, part-of-speech (POS) tagging and stemming.

For instance, in the sentence "Lung cancer risk among female textile workers exposed to endotoxin." (PubMed ID: 17341727), the NER full-text index recognizes the token "Lung cancer" as three distinct MeSH terms and two UMLS term (of the same "T191" type).

On the other hand, the linguistic index recognizes the verb stems of the tokens "Workers" and "exposed", for instance. The word stems are important in order to match declination or conjugated forms of the tokens in document retrieval and sentence retrieval.

The full-text indexes consume the most disk space in our database, for instance, 21.84 GB for PubMed titles, 152.75 GB for PubMed abstracts, 15.6 GB for PMC OA abstracts and 432.57 GB for PMC OA bodies. In comparison to this, indexing the MeSH terms description occupies only 0.0089 GB.

3.2 Question Processing

Our QA system classifies the questions in three types, namely: definition, factoid and summary. Definition questions, e.g., "What is zika virus?", expect a definition of a concept and are answered by our system with a definition from MeSH, if the term is found. Factoid questions, e.g., "Which are treatments for lung cancer?", output a short answer (more specifically, one or more MeSH terms), such as one or more treatments for a disease. If neither of these two types is recognized, the question is assigned the to the "summary" type, i.e., a short paragraph of text as answer. An example would be "What is the role of necroptosis in cancer therapy?".

Question Type Detection. This step determines the type of answer that should be returned. The system uses the following regular expressions: (a) factoid questions - ((list—name) .* ?) and ((what—where—which—who) (<plural noun>—are) .* ?); (b) definition questions - (what (is—are) <MeSH term> ?). All expressions are case insensitive.

Answer type detection. The detection of the answer type is carried out only for factoid questions and it is split in two parts: headword and type detection. The headword is(are) the word(s) which determines the type of answer, for instance, "symptoms" in the question "List common symptoms of patients with the DOORS syndrome." We rely on our NER full text index to detect the headwords. MeSH terms are preferred because the advantages of the MeSH tree structure can be used. If no MesH term is found, the system checks matches with the UMLS terminologies. The headword is always the first noun of the question, independent if considering MeSH or UMLS terms. In this step the system returns all possible types to which the answers can belong, either coming from a MeSH or a UMLS term. We have manually mapped the MeSH upper categories into the UMLS semantic types, to allow, for instance, answers coming from UMLS terms even if the detected headword was mapped to a MeSH term. In summary, the output of this component is a list of either MeSH ids (and the corresponding categories) or UMLS types.

Query formulation. In this step, the system converts the question into a query to the document retrieval component. We consider all relevant tokens from the sentences and, additionally, some meta information. For each token in the original question, the system can include its surface form or its MeSH or UMLS CUI identifier, if matches are found. We consider five categories, as illustrated in Figure 2, according to the following descending order of importance: (1) MeSH terms, (2) proper names, (3) nouns, (4) UMLS terms, and (5) adjectives/verbs/adverbs. If more than one match is found for a token, only the more important one is considered. The identification of proper names and the POS tags is obtained from the linguistic index (cf. above). Stopwords are removed from the query based on a stopword list of 319 English words retrieved from http://xpo6.com/list-of-english-stop-words/ (Accessed June 2016).

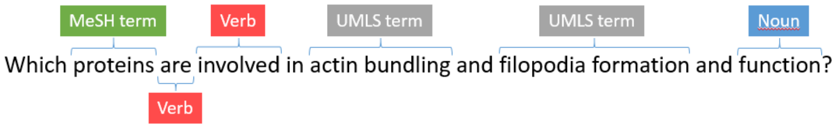
MeSH term
Verb
UMLS term
UMLS term
Noun
Which proteins are involved in actin bundling and filopodia formation and function?
Verb

Figure 2: Categories matched to the tokens of an input question.

3.3 Document Retrieval

The system narrows the number of documents that will go further in the workflow to the next components and from which the answer will be extracted. We utilize the built-in functionality of the database which allows fuzzy (approximate) search of keywords in the full text, such as linguistic variations of the word. For instance, the following sentence can be retrieved based on the keywords ”filopodia formation” and ”involve” (stem of the verb): ”HBXIP enhances the migration of breast cancer through increasing filopodia formation involving MEKK2/ERK1/2/Capn4 signaling.” (PMID 25304384).

Additionally, we have developed a simple algorithm to deal with situations in which no document could be found. The goal of the algorithm is to rank higher those documents which match the most important keywords. Firstly, the system performs a search with all keywords. If none is found, it removes the least important keyword, then the second least important keyword, and so on. The importance of each keyword is given by 2^N, where N is the number of occurrence of the keyword in the query.

3.4 Sentence retrieval

The system searches for relevant sentences that could contain the answer to the question, based on the documents retrieved in the previous step. We utilize specific scores for each of the five categories (cf. above). These score were obtained experimentally, namely: 1.7 for MeSH terms, 1.9 for proper names, 1.8 for nouns, 1.2 for UMLS terms and 1.6 for verbs, adjectives and adverbs. By summing up the individual scores, we obtain the final score for the passage. The top 20 sentences with highest scores are returned from this component.

3.5 Answer Processing

This step produces the output answers for the three question types. For the "definition" type, the system simply outputs the definition of the MeSH term referred in the question. In the case of summaries, these are generated based on our entity-based algorithms [2]. For factoid questions, we return the concepts which were found in the retrieved sentence and which match the semantic types as detected above. For instance, for the question “List common symptoms of patients with the DOORS syndrome.”, the MeSH terms "deafness" and "seizures" are candidates for answers and they belong to the semantic type "Signs and Symptoms".

References

[1] Plattner, H. (2013) A Course in In-Memory Data Management: The Inner Mechanics of In-Memory Databases, Springer, 1st edition.

[2] Schulze, F. and Neves, M. (2016) Entity-Supported Summarization of Biomedical Abstracts. In *Fifth Workshop on Building and Evaluating Resources for Biomedical Text Mining at the 26th International Conference on Computational Linguistics (Coling).*