HAWK@QALD5 – Trying to answer hybrid questions with various simple ranking techniques

Ricardo Usbeck[1] and Axel-Cyrille Ngonga Ngomo[1]

University of Leipzig, Germany
{usbeck,ngonga}@informatik.uni-leipzig.de

Abstract. The growing amount of data available in the Document Web as well as in the Linked Data Web has lead to an information gap. Information needed to answer complex questions might often require full-text data as well as Linked Data. Thus, HAWK combines unstructured and structured data sources.
In this article, we introduce HAWK, a novel entity search approach for hybrid question answering based on combining Linked Data and textual data. In this article, we compare three ranking mechanism and evaluate their performance on the QALD-5 challenge. Finally, we identify the weak points of our current version of HAWK and give directions for future development.

1 Introduction

The fifth challenge on Question Answering over Linked Data (QALD-5)[1] has introduced and extended a novel benchmark dataset for hybrid question answering. In this paper, we present our framework HAWK, the second best performing system w.r.t. hybrid question answering. The need for this challenge becomes obvious by comparing the growing amount of data, in the Document Web and the Linked Data Web, which introduces an information gap, i.e., a considerable number of complex questions can only be answered by using hybrid question answering approaches, which can find and combine information stored in both structured and textual data sources [4].

In this paper, we will outline the single steps performed by HAWK to answer hybrid questions in Section 2. Section 3 discusses problems and opportunities within the current implementation. Finally, we conclude in Section 4 and explain future research directions. The source code of HAWK, benchmark data, a link to the demo, evaluation results as well as further information can be retrieved from the project website `http://aksw.org/Projects/hawk`.

2 Method

In the following, we will shortly describe the 8-step, modular pipeline of HAWK absed on the following running example: `Which anti-apartheid activist was born in Mvezo?` For more information please have a look at the full method description [5].

[1] `http://www.sc.cit-ec.uni-bielefeld.de/qald/`

1. Segmentation and Part-of-Speech (POS)-Tagging. To be generic with respect to the language of the input question, HAWK uses a modular system that is able of tokenizing even languages without clear separation like Chinese. For English input questions our system relies on the *clearNLP* [1]-framework which provides a.o. a white space tokenizer, POS-tagger and transition-based dependency parsing. Afterwards, this framework annotates each token with its POS-tag which will be later used to identify possible semantic annotations. POS-tagging on the running example will result in the following: `Which(WDT) anti-apartheid(JJ) activist(NN) was(VBD) born(VBN) in(IN) Mvezo(NNP)?`

2. Entity Annotation. Next, our approach identifies named entities and tries to link them to the underlying knowledge base. The QALD-5 challenge relies on the DBpedia [2] as source for structured information in the form of Linked Data. For recognizing and linking named entities HAWK's default annotator is FOX [3], a federated knowledge extraction framework based on Ensemble Learning. FOX has shown to outperform other entity annotation systems on the QALD 4 benchmark data [5]. An optimal annotator would annotate our running example `Mvezo` with `http://dbpedia.org/resource/Mvezo`.

3. Dependency Parsing and Noun Phrase Detection. Subsequently, in order to capture linguistic and semantic relations, HAWK parses the query using dependency parsing and semantic role labeling [1]. The dependency parser generates a predicate-argument tree based on the preprocessed question. Afterwards, HAWK identifies noun phrases, i.e., semantically meaningful word groups not yet recognized by the entity annotation system, using the result of the POS tagging step. Input tokens are combined following manually-crafted linguistic heuristics based on POS-tag sequences derived from the QALD 5 benchmark questions. Two domain experts implemented the deduced POS-tag sequences and safeguarded the quality of this algorithm w.r.t. the QA pipeline f-measure. The full algorithm can be found in our source code repository at `https://github.com/aksw/hawk`. Here, the `anti-apartheid activist` would be detected as noun phrase.

4. Linguistic Pruning. HAWK has now captured entities from a given knowledge base, noun phrases as well as the semantic structure of the sentence. Still, this structure contains tokens that are meaningless for retrieving the target information or even introduce noise into the process. Thus, HAWK prunes nodes from the predicate-argument tree based on their POS-tags, e.g., deleting all `DET` nodes, interrogative phrases such as `Give me` or `List`, and auxiliary tokens such as `did`. The linguistically pruned dependency tree with combined noun phrases for our running example would only contain `born` as a root node with two children, namely `anti-apartheid activist` and `http://dbpedia.org/resource/Mvezo`.

5. Semantic Annotation. Now, the tree structure contains only semantically meaningful (combined) token and entities, i.e, individuals from the underlying knowledge base. To map the remaining token to properties and classes from the target knowledge base and its underlying ontology, our framework uses information about possible verbalizations of ontology concepts and leverages a fuzzy string search. These verbalizations are based on both `rdfs:label`[2] information from the ontology itself and (if available) verbalization information contained in lexica, in our case in the existing DBpedia English lexicon[3]. After this step, either a node is annotated with a class, property or individual from the target knowledge base or it causes a full-text lookup in the targeted Document Web parts. With respect to the running example `born` would be annotated with the properties `dbo:birthPlace` and `dbo:birthDate`.

6. Generating hybrid SPARQL Queries. Given a (partly) annotated predicate argument, HAWK generates hybrid SPARQL queries. It uses an Apache Jena FUSEKI[4] server, which implements the full-text search predicate `text:query` on a-priori defined literals. Those literals are basically mappings of textual information to a certain individual URI from a target knowledge, i.e., an implicit enrichment of structured knowledge from unstructured data.

Especially, our framework HAWK indexes a-priori the following information per individual: `dbo:abstract`, `rdfs:label`, `dbo:redirect` and `dc:subject` to capture document based information. This information is then retrieved by the `text:query` predicate by using exact matches or fuzzy matches on each non-stopword token of an indexed field.

The generation of SPARQL triple pattern is based on a pre-order walk to reflect the empirical observation that i) related information is situated close to each other in the tree and ii) information is more restrictive from left to right. This breadth-first search visits each node and generates several *possible triple patterns* based on the number of annotations and the POS-tag itself. With this approach HAWK is independent of SPARQL templates and to work on natural language input of any length and complexity. Each pattern contains at least one variable from a pre-defined set of variables, i.e., `?proj` for the resource projection variable, `?const` for resources covering constraints related to the projection variable as well as a variety of variables for predicates to inspect the surrounding of elements in the knowledge base graph. During this process, each iteration of the traversal appends the generated patterns to each of the already existing SPARQL queries. This combinatorial effort results in covering every possible SPARQL graph pattern given the predicate-argument tree. Amongst others, HAWK generates for the running example the following three hybrid SPARQL queries:

1. `SELECT ?proj {?proj text:query 'anti-apartheid activist'. ?proj dbo:birthPlace dbr:Mvezo.}`
2. `SELECT ?proj {?proj text:query 'anti-apartheid activist'. ?proj dbo:birthDate dbr:Mvezo.}`
3. `SELECT ?proj {?proj text:query 'anti-apartheid activist'. ?const dbo:birthPlace ?proj.}`

7. Semantic Pruning of SPARQL Queries. Covering each possible SPARQL graph pattern with the above algorithm results in a large number of generated SPARQL queries.

[2] We assume `dbo` stands for `http://dbpedia.org/ontology`, `dbr` for `http://dbpedia.org/resource/`, `rdfs` for `http://www.w3.org/2000/01/rdf-schema#`, `dc` for `http://purl.org/dc/elements/1.1/` and `text` for `http://jena.apache.org/text#`

[3] `https://github.com/cunger/lemon.dbpedia`

[4] `http://jena.apache.org/documentation/serving_data/`

To effectively handle this large set of queries and reduce the computational effort, HAWK implements various methods for pruning. For example, it assumes that unconnected query graphs, missing projection variables and cyclic SPARQL triple pattern lead to empty or semantically meaningless results. Thus, HAWK discards those queries.

In the running example, the semantic pruning discards query number two from above because it violates the range restriction of the `dbo:birthDate` predicate. Although semantic pruning drastically reduces the amount of queries, it often does not result in only one query. Thus, a ranking of the remaining queries is applied before the best SPARQL query is send to the target triple store.

8. Ranking and Cardinality. For the QALD-5 challenge, we extended the basic ranking implementations - optimal ranking and feature-based ranking - of HAWK by one additional ranking method:

– **Optimal Ranking.** To ensure, we are able to generate hybrid SPARQL queries capable of answering the benchmark questions, the optimal ranker returns always those hybrid SPARQL queries which lead to a maximum f-measure. Obviously, the optimal ranking can only be used if the answers are know, i.e., HAWK operates on the training data. This ranking functions allows to determine the parts of the hybrid question answering pipeline which do not perform well.

– **Feature-based Ranking.** The second ranking method is based on supervised learning using the gold standard answer set from the QALD-4 benchmark. In the *training phase*, all generated queries are run against the underlying SPARQL endpoint. Comparing the results to the gold standard answer set, HAWK stores all queries resulting with the highest F-measures. Afterwards the stored queries are used to calculate an average feature vector comprising simple features mimicking a centroid-based cosine ranking. HAWK's ranking calculation comprises the following components:
  - **NR_OF_TERMS** calculates the number of nodes used to form the full-text query part as described above.
  - **NR_OF_CONSTRAINTS** counts the amount of triple patterns per SPARQL query.
  - **NR_OF_TYPES** sums the amount of patterns of the form `?var rdf:type cls`.
  - **PREDICATES** generates a vector containing an entry for each predicate used in the SPARQL query.

  While running the *test phase* of HAWK, the cosine similarity between each SPARQL query using the above mentioned features and the average feature vector of training queries is calculated to rank them.

– **Overlap-based Ranking.** The novel overlap-based ranking accounts for the intuition that the same result set can be generated by several hybrid SPARQL queries. Thus, this ranker, although computationally highly expensive, executes every hybrid SPARQL query and the resulting answer sets are then stored into hashed buckets. Finally, the ranker computes how many queries produced a certain answer set. The answer set with the highest number is than returned.

Moreover, HAWK determines the target cardinality x, i.e., `LIMIT` x respectively the number of answers expected for a given query, of each query using the indicated cardinality of the first seen POS-tag, e.g., the POS-tag `NNS` demands the plural while `NN` demands the singular case and thus leads to different `x`.

An optimal ranking will reveal that the winning SPARQL query for our running example is `SELECT ?proj {?proj text:query 'anti-apartheid activist'. ?proj dbo:birthPlace dbr:Mvezo.}`.

3 Evaluation and Discussion

The QALD-5 benchmark has a training and a test dataset for question answering containing a subset of hybrid benchmark questions. In particular, HAWK is currently only capable of answering questions demanding a single or a set of URIs from a target knowledge base. Moreover, questions depending on Yago ontology[5] types cannot be answered. Thus, the QALD-5 dataset contains 26 training, respectively 8 test questions, suitable for the current implementation of HAWK. Using the online available evaluation tool[6], Table 1 shows the results for the training and test dataset as well as well as for all three ranking approaches. Please note, that for the feature-based ranker the training data was taken from QALD-4.

Table 1: Results of the QALD-5 challenge for different ranking algorithms. Number in brackets show the amount of generated answers, i.e., HAWK outputs at least one result set.

Dataset	Optimal Ranking	Feature-based Ranking	Overlap-based Ranking
QALD-5 - training	0.30 (15 out of 26)	0.06 (22 out of 26)	0.08 (22 out of 26)
QALD-5 - test	0.1 (1 out of 10)	0.10 (3 out of 10)	0.10 (3 out of 10)

As can be seen in Table 1, the implemented ranking function do not reach an optimal ranking implementation. Moreover, no implementation is able to discard whole queries, i.e., they are currently not aware of the possibility that the answer could not have been retrieved at all while the optimal ranker discards questions with incorrect answer sets.

Table 2 shows a detailed description of achieved measures per question and algorithm. Delving deeper into the results, we color-coded the single mistakes of the HAWK system. (Yellow) indicates that the ranking algorithms are not able to retrieve the correct SPARQL query out of the set of generated SPARQL queries. (Orange) cells point out that one ranking algorithm performs worse than the other. (Red) indicates the inability to retrieve correct answer sets. That is, HAWK is able to generate a set of SPARQL queries but amongst them none retrieves a correct answer set. Finally, (Blue) rows describe questions where HAWK is unable to generate at least one SPARQL query. Thus, those questions semantics cannot be captured by the system yet due to missing surface forms for individuals, classes and properties or missing indexed full-text information.

[5] `http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/`

[6] `http://greententacle.techfak.uni-bielefeld.de/~cunger/qald/index.php?x=evaltool&q=5`

Especially, in the test dataset all three ranking algorithms are only able to generate one correct answer while the feature-based and the overlap-based ranker perform differently on the train dataset. To experience the run times for single queries please visit our demo at `http://hawk.aksw.org`.

4 Conclusion

In this paper, we briefly introduced HAWK, the first hybrid QA system for the Web of Data, and analysed its performance against the QALD 5 challenge. We showed that by using a generic approach to generate SPARQL queries from predicate-argument structures, HAWK is able to achieve an F-measure of up to 0.3 on the QALD-5 training benchmark. These results demand novel approaches for capturing the semantic meanings of natural language questions with hybrid SPARQL queries. Our work on HAWK, however, revealed several other open research questions, of which the most important lies in finding the correct ranking approach to map a predicate-argument tree to a possible interpretation. So far, our experiments reveal that the mere finding of the right features for this endeavor remains a challenging problem. Finally, several components of the HAWK pipeline are computationally very complex. Finding more time-efficient algorithms for these steps will be addressed in future work.

Acknowledgments This work has been supported by the FP7 project GeoKnow (GA No. 318159) and the BMBF project SAKE.

References

1. J. D. Choi and M. Palmer. Getting the most out of transition-based dependency parsing. In *ACL*, pages 687–692, 2011.
2. J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer. DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, 2014.
3. R. Speck and A.-C. N. Ngomo. Ensemble learning for named entity recognition. In *ISWC*. 2014.
4. R. Usbeck. Combining Linked Data and Statistical Information Retrieval. In *11th ESWC, PhD Symposium*, 2014.
5. R. Usbeck, A.-C. Ngonga Ngomo, L. Bühmann, and C. Unger. HAWK - Hybrid Question Answering over Linked Data. In *ESWC*, 2015.

Table 2: Detailed results of HAWK at the QALD-5 challenge.

		Feature-based Ranking			Overlap-based Ranking			Optimal Ranking		
ID	**Question**	**Recall**	**Precision**	**F1**	**Recall**	**Precision**	**F1**	**Recall**	**Precision**	**F1**
301	Who was vice-president under the president who authorized atomic weapons against Japan during World War II?	0	0	0	0	0	0	1	1	1
303	Which anti-apartheid activist was born in Mvezo?	0	0	0	1	0.5	0.67	1	0.5	0.67
305	Which recipients of the Victoria Cross died in the Battle of Arnhem?	0	0	0	0.5	0.33	0.4	0.5	0.5	0.5
306	Where did the first man in space die?	0	0	0	0	0	0	1	1	1
308	Which members of the Wu-Tang Clan took their stage name from a movie?	0.5	0.08	0.14	1	0.17	0.29	0.5	0.5	0.5
309	Which writers had influenced the philosopher that refused a Nobel Prize?	0	0	0	0	0	0	1	1	1
311	Who composed the music for the film that depicts the early life of Jane Austin?	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
314	Which horses did The Long Fellow ride?	0	0	0	0	0	0	0.86	1	0.92
315	Of the people that died of radiation in Los Alamos, whose death was an accident?	0	0	0	0	0	0	0.5	1	0.67
316	Which building owned by the Bank of America was featured in the TV series MegaStructures?	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
317	Which buildings in art deco style did Shreve, Lamb and Harmon design?	1	0.33	0.5	0	0	0	1	0.33	0.5
318	Which birds are protected under the National Parks and Wildlife Act?	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67
319	Which country did the first known photographer of snowflakes come from?	0	0	0	0	0	0	1	1	1
320	List all the battles commanded by the lover of Cleopatra.	0	0	0	0	0	0	0.23	0.42	0.29
322	Which actress starring in the TV series Friends owns the production company Coquette Productions?	0	0	0	0	0	0	1	1	1
323	Gaborone is the capital of which country member of the African Union?	1	1	1	1	1	1	1	1	1
326	For which movie did the daughter of Francis Ford Coppola receive an Oscar?	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
327	Which city does the first person to climb all 14 eight-thousanders come from?	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
328	At which college did the only American actor that received the César Award study?	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
332	What is the native city of Hollywood's highest-paid actress?	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
333	In which city does the former main presenter of the Xposé girls live?	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
334	Who plays Phileas Fogg in the adaptation of Around the World in 80 Days directed by Buzz Kulik?	0	0	0	0	0	0	1	1	1
335	Who is the front man of the band that wrote Coffee & TV?	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
336	Which Chinese-speaking country is a former Portguese colony?	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
337	What is the largest city in the county in which Faulkner spent most of his life?	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
340	A landmark of which city is the home of the Mona Lisa?	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
51	Where was the "Father of Singapore" born?	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
52	Which Secretary of State was significantly involved in the United States' dominance of the Caribbean?	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
53	Who is the architect of the tallest building in Japan?	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
55	In which city where Charlie Chaplin's half brothers born?	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
56	Which German mathematicians were members of the von Braun rocket group?	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
57	Which writers converted to Islam?	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
59	Which movie by the Coen brothers stars John Turturro in the role of a New York City playwright?	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
60	Which of the volcanoes that erupted in 1550 is still active?	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦