CACAO: Conditional Spread Activation for Keyword Factual Query Interpretation

Edgard Marx[1,2,3](✉), Gustavo Correa Publio[1], and Thomas Riechert[1,2,3]

[1] LiberAI, AKSW, Leipzig University, Leipzig, Germany
{marx,gustavo.publio}@informatik.uni-leipzig.de
[2] HTWK, Leipzig, Germany
{edgard.marx,thomas.riechert}@htwk-leipzig.de
[3] Institute for Digital Technologies, Leipzig, Germany
http://liberai.org, http://ifdt.org/

Abstract. Information retrieval is regarded as pivotal to empower lay users to access the Web of Data. Over the past years, it achieved momentum with a large number of approaches being developed for different scenarios such as entity retrieval, question answering, and entity linking. This work copes with the problem of entity retrieval over RDF knowledge graphs using keyword factual queries. It discloses an approach that incorporates keyword graph structure dependencies through a conditional spread activation. Experimental evaluation on standard benchmarks demonstrates that the proposed method can improve the performance of current state-of-the-art entity retrieval approaches reasonably.

1 Introduction

Over the last years, information aplenty has been published as structured data. The *Resource Description Framework (RDF)*[1] became a standard format for many knowledge graphs (KG) publicly available such as DBpedia [18] and Wikidata [26]. An RDF KG organizes the information in the form of subject-predicate-object statements expressing semantic relations between entities (e.g. persons, organizations, and places) and concepts (e.g. given names, addresses, and locations). Currently, approximately 10.000 RDF KGs are available via public data portals.[2] Together, these graphs compose the so-called Linked Open Data Cloud (LOD).

Ultimately, approaches designed to retrieve or use KG's information has been getting substantial attention. Some of these approaches are Entity Retrieval (ER), Entity Linking (EL), Entity Disambiguation (ED), and Question Answering (QA). ER specifies a category of information retrieval (IR) whereas the result of a natural language search query is an entity or an entity's property rather than a document. ER methods play a fundamental role in IR on KGs. It enables lay

[1] http://www.w3.org/RDF.
[2] http://lodstats.aksw.org/.

M. Acosta et al. (Eds.): SEMANTiCS 2019, LNCS 11702, pp. 256–271, 2019.
https://doi.org/10.1007/978-3-030-33220-4_19

users to access KG's information as well as other approaches on performing EL [7], ED [24,36], and QA [10,33,35] tasks. Improving ER methods can have a substantial impact on the whole IR chain.

ER on RDF KG has peculiar characteristics that make it stand apart from standard document retrieval. The information in KG is structured in entities, attributes, classes, and their relationships. Exploring this structure makes ER a thriving research topic. Early approaches applied bag-of-word document retrieval techniques [4,8,38]. The research has been shifted to explore the KG entities and concepts relations in fields, the field retrieval models [5]. Late studies focus on evaluating the word sequence and property-type influence [2,21,41]. Recently, the use of EL is being considered for ER improvement [14].

This work presents `CACAO`, a novel approach for ER on large[3] and diverse RDF KGs. It relies on a novel spread activation (SA) method to improve information access. SA is a method that iteratively propagates weights in a graph from one node to another [6]. It differs from the previous approaches by evaluating query's intent on entities and concepts rather than fields and avoiding keyword over- and under-relatedness-estimation by accounting only the highly activated ones. The evaluation of the approach in two standard benchmarks shows an f-measure improvement of ≈10%.

The remaining of this work is organized as follows. Section 2 defines RDF KG and states the problem. Section 3 describes the conditional spread activation model entitled `CACAO`. Section 4 presents the evaluation and discusses the results. Section 5 provides a literature overview on related work. Finally, Sect. 6 concludes giving an outlook on approach limitations and potential future work.

2 Preliminaries

An RDF KG can be regarded as a set of triples in the form of $<s, p, o> \in (I \cup B) \times P \times (I \cup L \cup B)$ where: I is the set of all IRIs; B is the set of all blank nodes, $B \cap I = \emptyset$; P is the set of all predicates, $P \subseteq I$; E is the set of all entities, $E = I \cup B \setminus P$; L is the set of all literals, and; R is the set of all resources $R = I \cup B \cup P \cup L$. In this graph, an entity type is specified by the property `rdf:type` while the label, by the property `rdfs:label`. A field of an entity is a predicate object $f = <p, o>$ belonging to an entity triple $<e, p, o>$. The aim of entity retrieval is to recover the `top-K` ranked entities that best address the information need behind a given query as follows.

Definition 1 (Problem Statement). *Formally, a* `top-K` *entity retrieval takes a keyword query Q, an integer $0 < k$, a set of entities $E = \{e_1, e_2, ..., e_{|E|}\}$, and returns the top-k entities based on a scoring function $S(Q, e)$.*

3 The Approach

CACAO is an ER approach to facilitate information access using keyword factual queries in RDF knowledge graphs. Factual queries are those whose intent can

[3] We define large KGs as those having over a billion facts.

be formalized by simple Basic Graph Patterns (BGP).[4] Entity retrieval on KGs has been a long-studied research topic for many years. Early approaches rely on bag-of-words models [4,8,38] that suffers from *unrelatedness* [5] and *verbosity* [29]. They were built under the assumption that the distribution of keywords is proportional to its subject relatedness [19]. This idea contradicts with the fact that people can describe things differently. Authors can be more descriptive or verbose than others. Particularly in case of DBpedia, editors' experience or knowledge can unconsciously influence keyword frequency or even graph connectivity. To address the problem of verbosity, researchers proposed to score keywords normalized by the information (entity) length [29]. Other generation of ER approaches focused on the problem of unrelatedness by employing field retrieval models [5]. Late studies focused on evaluating how to weight fields differently so that to improve ER accuracy [2,21,41]. Nevertheless, field retrieval models are unable to relate query keywords with a specific predicate or object because they are treated as one, a bag-of-(field-words). Recent approaches introduced the use of two stage techniques employing ER followed by an Entity Link Retrieval (ELR) [14].

CACAO addresses the ER problem in a different manner. It relies on a SA method that works in threefold. A query triggers an activation function that measures the relatedness of KG resources w.r.t. the query. The resource relatedness values are then spread to their connected entities using a conditionally backward propagation, and, in a latter process, conditionally forward. The individual resource relatedness measurement addresses the problem of finding the query's intent. The conditional propagation avoids the over- and the underestimation of frequent and rare keywords. The next sections describes how the (1) Activation, (2) Conditional Backward Propagation and (3) Conditional Forward Propagation works.

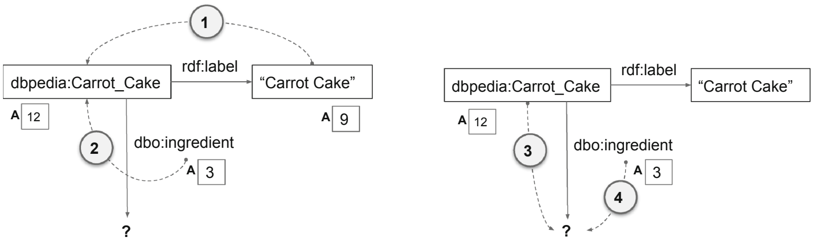
3.1 Activation

CACAO performs the activation in the resources. It uses the resource label coverage to evaluate its query relatedness. In this judgment, a query containing *birth date* should be more related to the property dbo:birthDate than to the property dbo:deathDate or dbpprop:date, while the query *date* should be more related with the property dbpprop:date than dbo:birthDate. Equation 1 formalizes the evaluation of the query label's coverage. It receives as parameters the query $\overrightarrow{Q}$ and a resource label $\overrightarrow{L}$ represented by bit vectors. In these vectors, keywords are dimensions in which their occurrence are either zero or one.

$$C(\overrightarrow{Q}, \overrightarrow{L}) = \frac{\sum \overrightarrow{Q}_i \overrightarrow{L}_i}{\sum \overrightarrow{L}_i} \tag{1}$$

Yet, the equation above cannot be used as an activation function, because it measures equally resources with the same query coverage rate. For the sake of

[4] For Basic Graph Pattern definition, visit http://www.w3.org/TR/rdf-sparql-query/#BasicGraphPatterns.



(a) The picture illustrates the conditional backward activation being performed on query "carrot cake ingredients". The activation value of the literal "Carrot Cake" and the property `dbo:`ingredient is being transfered to the entity `dbpedia:`Carrot_Cake (①-②).

(b) The picture illustrates the conditional forward activation being performed on query "carrot cake ingredients".The activation value of the entity `dbpedia:`Carrot_Cake and the property `dbo:`ingredient is being transfered to the property's entities (③-④).

Fig. 1. Conditional Spread Activation on query "carrot cake ingredients".

illustration, let us take as an example the query "carrot cake". For this query, either dbr:Carrot, dbr:Cake and dbr:Carrot_Cake are going to have the same coverage value of one, although dbr:Carrot_Cake has two overlapping keywords. Thus, full label-query overlaps are evaluated as the number of query keywords to the power of label keywords, $(\sum \overrightarrow{Q}_i)^{\sum \overrightarrow{L}_i}$. The incomplete overlaps are still considered, but treated with less importance. For those, the query-label intersects over their union suffices (Eq. 2). Equation 3 outlines the activation function. Notice, however, two important properties. First, entities whose resources were activated for mere casualty will always valuate lower than the query length $(\sum \overrightarrow{Q}_i)$. Second, it makes it easier to differentiate among resources with full and partial query coverage.

$$J(\overrightarrow{Q}, \overrightarrow{L}) = \frac{\sum \overrightarrow{Q}_i \overrightarrow{L}_i}{\sum \overrightarrow{L}_i + \sum \overrightarrow{Q}_i - \sum \overrightarrow{L}_i \overrightarrow{Q}_i} \tag{2}$$

$$A(\overrightarrow{Q}, \overrightarrow{L}) = \begin{cases} J(\overrightarrow{Q}, \overrightarrow{L}) & if\ C(\overrightarrow{Q}, \overrightarrow{L}) < 1; \\ (\sum \overrightarrow{Q}_i)^{\sum \overrightarrow{L}_i} & otherwise. \end{cases} \tag{3}$$

3.2 Conditional Backward Propagation

Backward Propagation consists of distributing backward computed values through a network. It is used in neural networks to transfer the errors throughout the network's layers [12]. In `CACAO`, the backward propagation is used to spread the resource's activation values to their connected entities. By doing so, the approach computes implicitly the relatedness of the entity and its connected resources to the query elements. However, the transfer is conditioned only to the most activated keyword value. It spreads from the resource to the fields, and, likewise, from the fields to the entity. This strategy prevents the frequency of the keywords on impacting the activation value while preserving their informativeness. For example, the entity dbpedia:Aristotle contains either dbo:birthDate and dbo:deathDate. In this case, the keyword "date" will have twice more impact on dbpedia:Aristotle then in entites containing solenly one of the properties (e.g. dbo:birthDate, dbo:deathDate or dbpprop:date).

Previous works demonstrate that scoring fields differently can improve the ER accuracy [2,21,41]. Hence, CACAO employs field weighting as described by Marx et al. [21]. Additionally, a query intent can be one or a set of entities. In the latter case, an important feature is the relevance ranking. As an example, the query "give me all persons" can return more than one million persons if applied to the DBpedia KG. But not all these entities may be relevant to the user. To deal with this problem, each activated entity receives a Page-Rank value normalized lower than a keyword weight. This work uses a modified version of *PageRank* [32] dubbed *DBpedia Page-Rank* which has been shown to produce better estimations [22].

Algorithm 1 describes the computation of the conditional backward propagation formally (Fig. 1a). It starts when function $A^f(\overrightarrow{Q}, f, \overrightarrow{L}^{\ominus})$ receives a bit vector representing the query $\overrightarrow{Q}$, the field, and a set of processed keywords. The activation field value (a_f) is initialized with 0, $R^f_{\blacktriangledown}$ with $\emptyset$, and R^f receives the field's resource list. In sequel, the function iterates over R^f computing the activation value a_f using the vectorized resource label returned by the function $\overrightarrow{V}^L(r)$. In line 19, the function INSERT operates an insertion sort on the list set $R^f_{\blacktriangledown}$. The insertion is performed in the ascending order of the resource's activation value to ensure that only the highly activated keywords have their value transfered to the entity. Subsequently, an iteration operates over the resource sorted list $R^f_{\blacktriangledown}$. The activation a_r is now evaluated over the resource label after removing the keywords that were computed on previous iterations $(\overrightarrow{L}^U_r)$. In the last iteration instructions, the resource activation value is transferred to the field a_f (line 25), and the resource keywords are added to the computed keyword list $\overrightarrow{L}^{\ominus}$ (line 26). The function resums adding the field's weight $\phi(f)$ to the final activation value a_f (line 28). Notice that we did not discuss the use of stop words removal or tokenization to describe the algorithm because they are optional and does not influence the overall computation.

The entity activation is computed over the fields' activation as follows. The function $A^r(\overrightarrow{Q}, e)$ receives a vectorized query $\overrightarrow{Q}$ and an entity e. The entity activation value a_e is initialized with 0. The computed keywords $\overrightarrow{L}^{\ominus}$ and the field set $F^e_{\blacktriangledown}$ receives $\emptyset$. The fieldset R^f receives the list of entity fields. Similar to the field activation function $A^f(\overrightarrow{Q}, f, \overrightarrow{L}^{\ominus})$, the entity activation consists in two iterations. The first (line 3) computes the field activation value a_f on every field's keyword, and uses an insertion sort function (line 5) to add them in $F^e_{\blacktriangledown}$ according to their inverse activation value. In this iteration, the computed keywords parameter $\overrightarrow{L}^{\ominus}$ from the field activation function $A^f(\overrightarrow{Q}, f, \overrightarrow{L}^{\ominus})$ receives an empty set (line `5`), allowing it to compute the activation on every keyword. It then iterates over the sorted fields $F^e_{\blacktriangledown}$ (line `8`) discarding the computed keywords, and transferring the field's activation value to the entity, a_e. The activation value then receives a normalized Page-Rank value returned by the $\psi(e)$ function.

3.3 Conditional Forward Propagation

The forward propagation is only applied when a property contributes to the field's activation. It forwards the entity activation to its activated properties, and from them to their objects. It results in objects having a higher activation value than their associated entity. Let us suppose that an user is looking for "carrot cake ingredients". In case of dbpedia:Carrot_Cake, the label activation will be backward propagated to the entity and then forwarded to the dbo:ingredient fields' object herewith the property activation. Thus, the dbo:ingredients' object on the BGP `<`dbpedia:Carrot_Cake dbo:ingredient `?object>` is going to have a higher activation value then dbpedia:Carrot_Cake. The Fig. 1b shows the conditional forward propagation for our running example query "carrot cake ingredients".

4 Evaluation

The evaluation was designed to measure the accuracy of `CACAO` compared to other ER, and Entity Linking methods. All output generated by the systems is publicly available at https://github.com/AKSW/irbench. There are several benchmark data sets that could be used on this task, including benchmarks from *Semantic Search* initiatives [13][5] and *QA Over Linked Data (QALD)*.[6] *Semantic Search* is based on user queries extracted from the Yahoo! search log, containing an average distribution of 2.2 words per-query. *QALD* provides both QA and keyword search benchmarks for RDF data. The QALD data sets are the most suitable due to the wide type of queries they contain and also because they make use of *DBpedia*, a very large and diverse KG. In this work, we use the QALD version 2 (QALD-2) data set benchmark from *The Test Collection for Entity Search* (`DBpedia-Entity`) [1], and; QALD version 4 (QALD-4) [34]. Table 1 shows the number of queries evaluated on each of them.

4.1 Experimental Setup

The evaluation contains two setups: The first setup evaluates `CACAO` against state-of-the-art Entity Retrieval (ER) using the QALD-2 from `DBpedia-Entity`. The second setup evaluates `CACAO` using state-of-the-art ER and Entity Linking Retrieval (ELR) for RDF data with the QALD-4. Both setups evaluate the approach with (`CACAO+F`) and without (`CACAO`) forward propagation.

[5] http://km.aifb.kit.edu/ws/semsearch10/.

[6] http://greententacle.techfak.uni-bielefeld.de/~cunger/qald/.

Table 1. Number of queries evaluated on each of the benchmark data sets.

Benchmark	#Queries
QALD-2 (`DBpedia-Entity`)	140
QALD-4	50

```
Data: $\overrightarrow{Q}$, the query. $e$, the entity.
Result: The entity final activation value $a_e$.
Function $A^e(\overrightarrow{Q}, e)$:
    $a_e \leftarrow 0$; $\overrightarrow{L}^{\ominus} \leftarrow \emptyset$; $F^e_{\blacktriangledown} \leftarrow \emptyset$; $F^e$, the fields in $< e, p, o >$ triples;
    forall $f \in F^e_{\blacktriangledown}$ do
        $a_f \leftarrow A^f(\overrightarrow{Q}, f, \overrightarrow{L}^{\ominus})$;
        $\overrightarrow{L}^{\ominus} \leftarrow \emptyset$;
        INSERT$(F^e_{\blacktriangledown}, f, a_f)$;
    end
    forall $f \in F^e_{\blacktriangledown}$ do
        $a_e \leftarrow a_e + A^f(\overrightarrow{Q}, f, \overrightarrow{L}^{\ominus})$;
    end
    $a_e \leftarrow a_e + \psi(e)$;
    return $a_e$;
end
Data: $f$, the field. $\overrightarrow{Q}$, the query. $\overrightarrow{L}^{\ominus}$, the vector containing the already
        computed keywords.
Result: The field score $a_f$.
Function $A^f(\overrightarrow{Q}, f, \overrightarrow{L}^{\ominus})$:
    $a_f \leftarrow 0$; $R^f_{\blacktriangledown} \leftarrow \emptyset$; $R^f$ the list of resources in $f$;
    forall $r \in R^f$ do
        $\overrightarrow{L}_r \leftarrow \overrightarrow{V}^L(r)$;
        $a_r \leftarrow A(\overrightarrow{Q}, \overrightarrow{L}_r)$;
        INSERT$(R^f_{\blacktriangledown}, r, a_r)$;
    end
    forall $r \in R^f_{\blacktriangledown}$ do
        $\overrightarrow{L}^r \leftarrow \overrightarrow{V}^L(r)$;
        $\overrightarrow{L}^U_r \leftarrow \overrightarrow{L}_r \wedge \neg \overrightarrow{L}^{\ominus}$;
        $a_r \leftarrow A(\overrightarrow{Q}, \overrightarrow{L}^U_r)$;
        $a_f \leftarrow a_f + a_r$;
        $\overrightarrow{L}^{\ominus} \leftarrow \overrightarrow{L}^{\ominus} \vee \overrightarrow{L}^U_r$;
    end
    $a_f \leftarrow a_f + \phi(f)$;
    return $a_f$;
end
```

Algorithm 1. A Conditional Backward Propagation.

First Setup. The first setup evaluates CACAO against thirteen different ER models distributed over three groups (Unstructured, Fielded and Other models) using the QALD-2 DBpedia-Entity data set benchmark. Results are reported using the benchmark standard evaluation metrics: Mean Average Precision (MAP) and Precision at rank 10 (P@10) [20]. The evaluated unstructured retrieval models use flattened entity representation: LM (Language Modeling) [27]; SDM (Sequential Dependence Model) [23], and; BM25 [29]. Five retrieval models employed fielded entity representation: MLM (Mixture of Language Models) [25]; FSDM (Fielded Sequential Dependence Model) [41]; BM25F [5]; MLM-all, with equal field weights, and; PRMS (Probabilistic Model for Semistructured Data) [15]. The LTR (Learning-to-Rank) approach [3] employs 25 features from various retrieval models trained using the RankSVM algorithm. All EL (Entity Link) methods used TAGME [11] for annotating queries with entities, and an URI-only index (with a single catchall field) for computing the EL component. CA suffixes refer to models that are trained using Coordinate Ascent.

Second Setup. The second setup extends the evaluation to QALD-4 benchmark on ER and EL. It measures the performance of eight different Levenshtein, Jaccard, BMF25F and CACAO baseline scoring functions: $Levenshtein_a$ uses the number of matched characters for each matched keyword; $Levenshtein_b$ uses the number of matched characters with the paraphrase disambiguation method proposed by Zhang et al. [40]; $Jaccard_a$ uses the Jaccard distance of matched resources per matched keyword; $Jaccard_b$ uses the disambiguating model implemented by Shekarpour et al. [31]; BMF25F is the ER method proposed by Blanco et al. [2], and; $\texttt{CACAO}_{P65}$ is the CACAO disambiguation model with rule 65 proposed by Shaekapour et al. [31][7]—$J(\overrightarrow{Q}, \overrightarrow{L}) \in [0.65, 1]$—applied only to properties. The idea is that there is a need to address inflections only on properties where verbs occur rather than objects that usually contain proper names. The $Levenshtein_a$ and $Jaccard_a$ methods are used to measure local keyword frequency without global occurrency normalization. CACAO and $\text{Glimmer}_{\text{Y!}}$ [2] performed all queries in OR mode. The performance considers only the *top-K* entries returned by each approach, where k equals to the number of entries in the target test query. The EL evaluation on Table 5 evaluates the mentioned baseline functions as well as the last version of DBpedia Spotlight (version 1.0), AGDISTIS [36] and the state-of-the-art ED MAG [24] in simply BGP queries. This evaluation was designed to measure how accurate CACAO can be when dealing with approaches that use EL on factual keyword queries. We discard queries that can only be answered using classes and properties. We avoid the use of these queries because annotators usually can only handle entities. QALD-4 has ten queries that follow this criteria, Queries 12, 13, 21, 26, 30, 32, 34, 41, 42, and 44. All queries evaluated over DBpedia Spotlight used a refinement operator approach starting from confidence 0.5 in decreasing scale of 0.05 until reaching an annotation—when it was possible—or zero. AGDISTIS [36] and MAG [24] were evaluated over manually marked entity queries.

[7] The optimal value is a range belonging to the interval [0.6, 0.7].

Table 2. Mean Average Precision (**MAP**) achieved by different Entity Retrieval models on QALD-2 *DBpedia-Entity* benchmark data set.

Approach	**MAP**
CACAO	**0.2417**
BM25-CA	0.1939
SDM+EL	0.1887
FSDM+EL	0.1719
BM25F-CA	0.1671
LTR	0.1629
LM+EL	0.1534
SDM	0.1533
LM	0.1424
FSDM	0.1403
MLM-CA	0.1273
PRMS	0.1103
BM25	0.1092
MLM-all	0.1062

Table 3. Precision 10 (**P@10**) achieved by different Entity Retrieval models on QALD-2 *DBpedia-Entity* benchmark data set.

Approach	**P@10**
CACAO	**0.3057**
BM25-CA	0.2527
LM+EL	0.2362
SDM+EL	0.2249
LM	0.2144
FSDM+EL	0.2113
BM25F-CA	0.2053
FSDM	0.2000
SDM	0.1883
PRMS	0.1871
MLM-CA	0.1844
MLM-all	0.1843
LTR	0.1732
BM25	0.0986

Table 4. ***P**recision*, ***R**ecall* and ***F**$_1$-measure* achieved by different ER approaches on QALD-4 benchmark data set.

Approach	**P**	**R**	**F$_1$**
CACAO+F	**0.19**	**0.19**	**0.19**
CACAO	0.11	0.11	0.11
CACAO$_{P65}$	0.09	0.09	0.09
Levenshtein$_b$	0.04	0.05	0.04
BM25F [2]	0.03	0.03	0.03
Jaccard$_b$	0.01	0.04	0.01
Levenshtein$_a$	0.00	0.00	0.00
Jaccard$_a$	0.00	0.00	0.00

Table 5. ***P**recision*, ***R**ecall* and ***F**$_1$-measure* achieved by different EL approaches on QALD 4 benchmark data set.

Approach	**P**	**R**	**F$_1$**
CACAO$_{P65}$	**1**	**1**	**1**
CACAO	0.90	0.90	0.90
MAG [24]	0.80	0.80	0.80
DBpedia Spotlight [7]	0.70	0.70	0.70
Levenshtein$_b$	0.60	0.60	0.60
Jaccard$_b$	0.60	0.60	0.60
AGDISTIS [36]	0.30	0.30	0.30
BM25F [2]	0.30	0.30	0.30
Levenshtein$_a$	0.00	0.00	0.00
Jaccard$_a$	0.00	0.00	0.00

Query and Resource Parsing. All implemented models (CACAO, CACAO+F, Jaccard and Levenshtein) perform the query and resource parsing extracting individual keywords, removing punctuation and capitalization as well as applying lemmatization.

4.2 Results

The results show that CACAO outperforms the state-of-the-art in both ER and EL tasks with keyword factual queries. It achieved ≈10% more accuracy than ER and EL approaches. Further, as expected, annotators performed better than ER on EL task. Tables 2 and 3 shows resp. the MAP and P@10 performance of CACAO compared to 13 methods. The tables show the score with a precision of four digits. It is possible to notice that MAP@10 scores considerably lower than P@10. That occurs because MAP is calculated on the average entry's precision per question while P is computed only over matching entries. It means that although the entities are retrieved, their query rank can still be improved. Except for CACAO, some methods achieved different position in P@10 and MAP. The outcomes reveal that CACAO could produce more (1) precise and (2) complete results. In general, except SDM, the results confirm previous findings [14] that shows that CA and EL approaches could achieve better performance than their simple version—without—while EL versioned methods performed better than CA ones. CACAO could outperform previous methods because it acts as a resource linking approach. It evaluates resource dependencies rather than bi and trigrams keyword dependencies used in fielded approaches. It also suppresses SDM weakness of sorting entities in relevance order [41] using Page-Rank.

Table 4 shows the Precision, Recall and F-measure achieved by each baseline models on QALD-4. CACAO achieved a better F-measure than CACAO$_{P65}$ mainly because it could overcome the problem of vocabulary mismatch on Query 29 by annotating the keyword "Australian" with dbpedia:Australia, and Query 49, by annotating the keyword "Swedish" with dbpedia:Sweden. As expected, methods empowered by disambiguation ($Levenshtein_a$ and $Jaccard_a$) scores better than bag-of-words ($Levenshtein_b$ and $Jaccard_b$). $Levenshtein_a$ scores better than $Jaccard_a$, confirming previous research conclusion [40]. However, $Jaccard_b$ and $Levenshtein_b$ have their major drawbacks in the path disambiguation level. When retrieval scoring functions consider keywords equally weighted, they cannot disambiguate among resources containing the same keywords. For instance, in case an user query "places", both property dbo:place and the entity-type dbo:Place can be equally weighted, leading these models to retrieve either places as well as the entities connected to the property dbo:place. Not surprisingly, there was an issue related to the local[8] term frequency on BMF25F [2] model. On Query 30, it retrieves the entity dbpedia:Halloween_(Dave_Matthews_Band_song) because the word "halloween" occurs more frequently than in the desired one (dbpedia:Halloween).

[8] Not to confuse with global term frequency.

Table 5 shows the EL evaluation over ten queries. There, $\texttt{CACAO}_{P65}$ achieved the highest F-measure of 1. CACAO achieved an F-measure of 0.90, obtaining ≈0.10% more accuracy than MAG, the third best-performing approach. CACAO annotates wrongly Query 21 keyword bach by dbpedia:Bachs. $\texttt{CACAO}_{P65}$ applied 65 rule only to the properties, assigning correctly dbpedia:Bach. MAG could not annotate correctly Query 34 and 44, and; DBpedia Spotlight Queries 12, 41, and 42. The results expose a deficiency of EL systems in dealing with single entity factual queries.

Entity Linking and Disambiguation approaches [7,24] exploit IR for finding the corresponding entity. For these systems, incomplete labels can lead to a non or an inconsistent annotation. For example, in our evaluation DBpedia Spotlight links the keyword "baldwin" in Query 47 with the entity dbpedia:Baldwin_Locomotive_Works. Other queries do not generate any annotation. That is the case of Query 36 whereas DBpedia Spotlight does not annotate it using confidence score 0.5, but annotates it wrongly using confidence 0.45.[9] The use of the 65 rule, enhanced the results achieved by CACAO when applied to subjects, properties, and objects in comparison to when applied to only properties ($\texttt{CACAO}_{P65}$), see Table 4. This happens because it can help to annotate noun resources that are not handled by the lemmatization, i.e., Sweden and Swedish on Query 43. However, the use of this method decreases the precision of the approach in Entity Linking task (see Table 5) because the 65 rule increases the possible overlaping resources leading to wrong annotations. That's the case of Query 21.

Complexity Analysis. In general, entity (document) retrieval algorithms can be implemented as an entity- or term-a-time. Entity-a-time retrieval algorithms aggregates scores over entities whereas term-a-time over terms. Term-a-time is the most common retrieval method and relies on posting lists implemented in popular IR frameworks such as Lucene. Intuitively, the complexity of term a time methods are bounded by the size of the posting list matching terms M' and E' matching entities insertions on a tree of size k (top-k) which leads to a complexity of $\mathrm{O}(M' + E' \log k)$.[10] Algorithm 1 display a naive implementation of our proposed entity-a-time method. The second For instruction (line 21) is bounded by the same time complexity of the number of the entity's matched terms, giving an overall collection complexity of $\mathrm{O}(M')$. However, when considering the first loop (line 16), there is a need for calculating the activation value on every entity's matched term, adding an extra complexity of matched term frequencies Tf'. Thus, the complexity of Algorithm 1 is at least $\Omega(Tf' + M' + E' \log k)$, highlighting a future point of improvement.

5 Related Work

IR. Existing IR approaches commonly aim to retrieve the *top-K* ranked documents for a given NL input query. Term Frequency-Inverse Document Frequency

[9] With confidence 0.45 "pope" is annotated with the entity dbpedia:Pope.

[10] We ignored the existence of fields and resources for simplification.

(TF-IDF) [30] evaluates query keywords based on their local and global frequency. BM25 [28] extends TF-IDF introducing a document length normalization. Field-base extensions from bag-of-words have been proposed for IR on structured data. BM25F [5] is an extension of BM25 to retrieve structured data using different weighted fields. Mixture of Language Models (MLM) [25] extends the Language Model (LM) [27] using a linear combination of query keyword probability in a multi-field language model (MLM). Although individual field weights in BM25F and MLM can be tuned for a particular collection, they are fixed across different query keywords. Probabilistic Retrieval Model for Semistructured Data (PRMS) [16] overcomes this limitation using a probabilistic classification to map query keywords into fields. Other IR approaches extend field retrieval models adding keyword dependencies. The Markov Random Field (MRF) retrieval model [23] proposes three variants of keyword query dependencies: (1) full independence (FIM); (2) sequential dependence (SDM), and; full dependence (FDM). Zhiltsov et al. [41] proposed an fielded ER model based on unigrams and bigrams applied to five different fields (names, categories, similar entity names, related entity names, and other attributes). The model uses different field weights for ordered (e.g., keywords that appear consecutive ly) and unordered bigrams. Koumenides et al. Hasibi et al. [14] shows that entity linking can improve entity retrieval models. Asi et al. [17] gives a comprehensive overview of ER approaches.

Semantic Web. Swoogle [9] introduces a modified version of PageRank that takes into consideration the types of the links between ontologies. Semplore [39], Falcons [4], and Sindice [8] explore traditional document retrieval for querying RDF data. YAHOO! BNC and Umass [13] were respectively the best and second best ER in SemanticSearch'10. YAHOO! BNC uses BM25F aplaying specific boosts on different fields (title, name, dbo:title, others). Blanco et al. [2] uses BM25F boosting important and unimportant fields differently. The proposed adaptation is implemented in the Glimmer$_{\mathrm{Y!}}$ engine and is shown to outperform other state-of-the-art methods on the task of ER. Virgilio et al. [37] introduced a distributed technique for ER on RDF data using MapReduce. The retrieval is carried out using only the high ranked (Linear) and all matched fields (Monotonic) strategies. Our work distinguish from the previous by (1) computing the similarity on the individual resources and avoiding the over- and the under-estimation of frequent and rare keywords.

6 Conclusion, Limitations and Future Work.

Whereas recent ER systems gain more precision, retrieving the desired information still imposes a major challenge. This work presented a conditional activation approach for efficient ER over RDF KG using factual query interpretation. The results show a significant improvement of accuracy in comparison to the state-of-the-art ER and EL systems in standard benchmark data sets. In particular, CACAO shows an increase of ≈10% on P@10 and MAP in standard ER benchmark data set. CACAO could outperform other ER and EL methods because it relies on a model that combines two properties: (1) It is a resource-based rather than a fielded retrieval approach, and; (2) It performs a conditional activation that avoids the over- and the under-estimation of frequent and rare keywords.

Nevertheless, there are a few challenges not addressed in the current implementation such as the keyword and character position as well as approach memory and runtime optimizations. Queries such as "peace and war" and "war and peace" can be activated equally. However, one can refer to dbpedia:Peace_and_War whereas the other to dbpedia:War_and_Peace. Recent works [41] have shown promising results in addressing this problem. The evaluation shows that current benchmarks do not address this issue. In future work, we plan to overcome the mentioned challenges. We see this work as the first step of a broader research agenda for designing more accurate ER systems over Linked Data.

Acknowledgments. This work was partly supported by German Research Foundation, project *Professorial Career Patterns of the Early Modern History* grant number 317044652 and CNPq Foundation under (200527/2012-6) and (201808/2015-3).

References

1. Balog, K., Neumayer, R.: A test collection for entity search in DBpedia. In: Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2013, pp. 737–740. ACM, New York (2013)
2. Blanco, R., Mika, P., Vigna, S.: Effective and efficient entity search in RDF data. In: Aroyo, L., et al. (eds.) ISWC 2011. LNCS, vol. 7031, pp. 83–97. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25073-6_6
3. Chen, J., Xiong, C., Callan, J.: An empirical study of learning to rank for entity search. In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2016, pp. 737–740. ACM, New York (2016)
4. Cheng, G., Qu, Y.: Searching linked objects with falcons: approach, implementation and evaluation. Int. J. Semant. Web Inf. Syst. **5**(3), 49–70 (2009)
5. Craswell, N., Zaragoza, H., Robertson, S.: Microsoft Cambridge at TREC 14: enterprise track. In: Voorhees, E.M., Buckland, L.P. (eds.) TREC, volume Special Publication 500-266. National Institute of Standards and Technology (NIST) (2005)
6. Crestani, F.: Application of spreading activation techniques in information retrieval. Artif. Intell. Rev. **11**(6), 453–482 (1997)
7. Daiber, J., Jakob, M., Hokamp, C., Mendes, P.N.: Improving efficiency and accuracy in multilingual entity extraction. In: Proceedings of the 9th International Conference on Semantic Systems, I-SEMANTICS 2013, pp. 121–124. ACM, New York (2013)
8. Delbru, R., Campinas, S., Tummarello, G.: Searching web data: an entity retrieval and high-performance indexing model. Web Semant. Sci., Serv. Agents World Wide Web **10**, 33–58 (2012)
9. Ding, L., et al.: Swoogle: a search and metadata engine for the semantic web. In: Proceedings of the Thirteenth ACM Conference on Information and Knowledge Management (CIKM), pp. 652–659. ACM (2004)

10. Dubey, M., Dasgupta, S., Sharma, A., Höffner, K., Lehmann, J.: AskNow: a framework for natural language query formalization in SPARQL. In: Sack, H., Blomqvist, E., d'Aquin, M., Ghidini, C., Ponzetto, S.P., Lange, C. (eds.) ESWC 2016. LNCS, vol. 9678, pp. 300–316. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-34129-3_19
11. Ferragina, P., Scaiella, U.: TAGME: on-the-fly annotation of short text fragments (by Wikipedia entities). In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, CIKM 2010, pp. 1625–1628. ACM, New York (2010)
12. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016)
13. Halpin, H., et al.: Evaluating ad-hoc object retrieval. In: Proceedings of the International Workshop on Evaluation of Semantic Technologies (IWEST 2010), Shanghai, PR China, 11 2010. 9th International Semantic Web Conference (ISWC2010) (2010)
14. Hasibi, F., Balog, K., Bratsberg, S.E.: Exploiting entity linking in queries for entity retrieval. In: Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval, pp. 209–218. ACM (2016)
15. Hasibi, F., Balog, K., Bratsberg, S.E.: Entity linking in queries: efficiency vs. effectiveness. In: Jose, J.M., et al. (eds.) ECIR 2017. LNCS, vol. 10193, pp. 40–53. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56608-5_4
16. Hasibi, F., et al.: DBpedia-entity V2: a test collection for entity search. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2017, pp. 1265–1268. ACM, New York (2017)
17. Koumenides, C.L., Shadbolt, N.R.: Ranking methods for entity-oriented semantic web search. J. Assoc. Inf. Sci. Technol. **65**(6), 1091–1106 (2014)
18. Lehmann, J., et al.: DBpedia - a large-scale, multilingual knowledge base extracted from Wikipedia. Semant. Web J. **6**(2), 167–195 (2015)
19. Luhn, H.P.: A statistical approach to mechanized encoding and searching of literary information. IBM J. Res. Dev. **1**(4), 309–317 (1957)
20. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, New York (2008)
21. Marx, E., Höffner, K., Shekarpour, S., Ngomo, A.-C.N., Lehmann, J., Auer, S.: Exploring term networks for semantic search over RDF knowledge graphs. In: Garoufallou, E., Subirats Coll, I., Stellato, A., Greenberg, J. (eds.) MTSR 2016. CCIS, vol. 672, pp. 249–261. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49157-8_22
22. Marx, E., Zaveri, A., Moussallem, D., Rautenberg, S.: DBtrends: exploring query logs for ranking RDF data. In: Proceedings of the 12th International Conference on Semantic Systems, SEMANTiCS 2016, pp. 9–16. ACM, New York (2016)
23. Metzler, D., Croft, W.B.: A Markov random field model for term dependencies. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2005, pp. 472–479. ACM, New York (2005)
24. Moussallem, D., Usbeck, R. , Röder, M., Ngonga Ngomo, A.-C.: MAG: a multilingual, knowledge-base agnostic and deterministic entity linking approach. In: K-CAP 2017: Knowledge Capture Conference, p. 8. ACM (2017)

25. Ogilvie, P., Callan, J.: Combining document representations for known-item search. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, SIGIR 2003, pp. 143–150. ACM, New York (2003)
26. Pellissier Tanon, T., Vrandečić, D., Schaffert, S., Steiner, T., Pintscher, L.: From freebase to Wikidata: the great migration. In: Proceedings of the 25th International Conference on World Wide Web, pp. 1419–1428. International World Wide Web Conferences Steering Committee (2016)
27. Ponte, J.M., Croft, W.B.: A language modeling approach to information retrieval. In: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1998, pp. 275–281. ACM, New York (1998)
28. Robertson, S., Zaragoza, H.: The probabilistic relevance framework: Bm25 and beyond. Found. Trends® Inf. Retr. **3**(4), 333–389 (2009)
29. Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M.M., Gatford, M., et al.: Okapi at trec-3. Nist Spec. Publ. Sp **109**, 109 (1995)
30. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Commun. ACM **18**(11), 613–620 (1975)
31. Shekarpour, S., Marx, E., Ngomo, A.-C.N., Auer, S.: SINA: semantic interpretation of user queries for question answering on interlinked data. J. Web Semant. **30**, 39–51 (2015)
32. Thalhammer, A., Rettinger, A.: Browsing DBpedia entities with summaries. In: Presutti, V., Blomqvist, E., Troncy, R., Sack, H., Papadakis, I., Tordai, A. (eds.) ESWC 2014. LNCS, vol. 8798, pp. 511–515. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11955-7_76
33. Unger, C., Bühmann, L., Lehmann, J., Ngonga Ngomo, A.-C., Gerber, D., Cimiano, P.: Template-based question answering over RDF data. In: Proceedings of the 21st International Conference on World Wide Web, WWW 2012, pp. 639–648. ACM, New York (2012)
34. Unger, C., et al.: Question answering over linked data (QALD-4). In: Cappellato, L., Ferro, N., Halvey, M., Kraaij, W. (eds.) Working Notes for CLEF 2014 Conference, Sheffield, United Kingdom (2014)
35. Usbeck, R., Ngonga Ngomo, A.-C., Bühmann, L., Unger, C.: HAWK – hybrid question answering using linked data. In: Gandon, F., Sabou, M., Sack, H., d'Amato, C., Cudré-Mauroux, P., Zimmermann, A. (eds.) ESWC 2015. LNCS, vol. 9088, pp. 353–368. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-18818-8_22
36. Usbeck, R., et al.: AGDISTIS - graph-based disambiguation of named entities using linked data. In: Mika, P., et al. (eds.) ISWC 2014. LNCS, vol. 8796, pp. 457–471. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11964-9_29
37. De Virgilio, R., Maccioni, A.: Distributed keyword search over RDF via MapReduce. In: Presutti, V., d'Amato, C., Gandon, F., d'Aquin, M., Staab, S., Tordai, A. (eds.) ESWC 2014. LNCS, vol. 8465, pp. 208–223. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07443-6_15
38. Wang, H., et al.: Semplore: a scalable IR approach to search the web of data. J. Web Semant. **7**(3), 177–188 (2009)
39. Zhang, L., Liu, Q.L., Zhang, J., Wang, H.F., Pan, Y., Yu, Y.: Semplore: an IR approach to scalable hybrid query of semantic web data. In: Aberer, K., et al. (eds.) ASWC/ISWC -2007. LNCS, vol. 4825, pp. 652–665. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76298-0_47

40. Zhang, Y., He, S., Liu, K., Zhao, J.: A joint model for question answering over multiple knowledge bases. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI 2016, pp. 3094–3100. AAAI Press (2016)
41. Zhiltsov, N., Kotov, A., Nikolaev, F.: Fielded sequential dependence model for ad-hoc entity retrieval in the web of data. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2015, pp. 253–262. ACM, New York (2015)