ITC 2/49
Information Technology and Control
Vol. 49 / No. 2 / 2020
pp. 275-288
DOI 10.5755/j01.itc.49.2.24985

Applying Semantic Role Labeling and Spreading Activation Techniques for Semantic Information Retrieval

Received 2019/12/30 | Accepted after revision 2020/05/06

http://dx.doi.org/10.5755/j01.itc.49.2.24985

HOW TO CITE: Vileniškis, T., Butkienė, R. (2020). Applying Semantic Role Labeling and Spreading Activation Techniques for Semantic Information Retrieval. *Information Technology and Control*, 49(1), 275-288. https://doi.org/10.5755/j01.itc.49.2.24985

Applying Semantic Role Labeling and Spreading Activation Techniques for Semantic Information Retrieval

Tomas Vileniškis, Rita Butkienė

Faculty of Informatics; Kaunas University of Technology;
Studentų Str. 50, LT-51368, Kaunas, Lithuania; e-mails: tomas.vileiniskis@ktu.lt, rita.butkiene@ktu.lt

Corresponding author: tomas.vileiniskis@ktu.lt

Semantically enhanced information retrieval is aimed at improving classical information retrieval methods and goes way beyond plain Boolean keyword matching with the main goal of better serving implicit and ambiguous information needs. As a de-facto pre-requisite to semantic information retrieval, different information extraction techniques are used to mine unstructured text for underlying knowledge. In this paper, we present a method that combines both information extraction and information retrieval to enable semantic search in natural language texts. First, we apply semantic role labeling to automatically extract event-oriented information found in natural language texts to a Resource Description Framework knowledge graph leveraging semantic web technology. Second, we investigate how a custom flavored graph traversal spreading activation algorithm can be employed to interpret user's information needs on top of the previously extracted knowledge base. Finally, we present an assessment on the applicability of our method for semantically enhanced information retrieval. An experimental evaluation on partial WikiQA dataset shows the strengths of our approach and also unveils common pitfalls that we use as guidelines to draw further work directions in the open-domain semantic search field.

KEYWORDS: Semantic role labeling, spreading activation, information retrieval, information extraction, ontologies, RDF.

1. Introduction

In the context of traditional web search, information retrieval (IR) has been known as a task of obtaining documents relevant to user's information needs, typically expressed by a form of a query. The precision of search results highly depends on two main criteria: translation of user queries and document content description. The nature of the web, however, imposes many challenges on web IR, one of them being a continuous growth of information available online. As the target search space increases, more focus should be directed towards effective document content processing in order to distinguish between new and repeated knowledge sources. Here, we encounter another paradigm known as information extraction (IE). IE may be seen as an activity of automatically extracting structured information from an unstructured or semi-structured information source. While IR and IE seem to be aimed at different tasks, a strong correlation exists between them [11] – we see structured content as the foundation for more precise IR.

Among many of the information extraction methods, semantic role labeling (SRL) has been gaining a lot of attention over the last couple of years. SRL is a task in natural language processing (NLP) that aims to parse natural language sentences into predicate-argument structures. In other words, SRL assigns predefined semantic roles to syntactic constituents of a sentence. Two of the most widely adopted resources for SRL are FrameNet [2] and PropBank [20]. The latter corpus is mostly used as a gold standard for automatic SRL, which is usually based on machine learning methods [4, 5]. Given the sentence D_1: „In November 2006, YouTube was bought by Google for US$1.65 billion, and operates as a subsidiary of Google.", its predicate-argument structure looks like the following:

P1: [A0: by Google] [V: buy.01] [A1: YouTube] [AM-TMP: In November 2006]

P2: [A0: YouTube] [V: operate.01] [A3: as a subsidiary of Google].

Such structure represents shallow semantics of a sentence where each of the predicates is accompanied by its main (A0, A1, A2) and adjunctive arguments (AM-TMP, AM-LOC, AM- MN). Since argument roles in SRL are determined on top of syntactic parses, this allows distinguishing between repeated and unique underlying knowledge even when it is expressed by using different syntactic variations in a sentence. Let us take a look at another sample sentence D_2: „In November 2006, Google bought YouTube for US$1.65 billion, which operates as a subsidiary of Google.", its predicate-argument structure looks like the following (only single verb buy.01 considered):

P1: [A0: Google] [V: buy.01] [A1: YouTube] [AM- TMP: In November 2006].

As can be seen from above, SRL extraction for D_2 results in the same predicate argument structure for the predicate buy.01 as in D_1, despite having an active voice construction as opposed to passive voice in D_1.

In addition to being able to capture shallow sentence meaning in a syntax-independent way, SRL outweighs other IE methods by drawing clear boundaries between core information bits (main arguments) and pure noisy words that play no semantic role in a sentence.

Most efforts and research to date have been devoted to methods for automatic labeling of semantic roles, while application of the resulting predicate-argument structures for IR purposes remains quite open and limited mostly to Question Answering (QA) systems [19, 21].

Therefore, in this paper we focus on application of predicate-argument structures for a more global IR task. We see SRL as a foundation to construct unique event-driven knowledge assertions. Since the natural ambiguity behind user's information needs and information sources cannot be covered by solely relying on shallow predicate argument structures, deep semantic analysis of the resulting arguments is necessary to be carried out. We employ ontological semantic analysis via DBpedia as a means to both, (1) disambiguating subject and object roles of a predicate to a knowledge base entity, and (2), using the typing information of entities for query expansion purposes. For example, the A0 and A1 arguments in P1 (D_1, D_2) map to DBpedia entries http://dbpedia.org/resource/Google and http://dbpedia.org/resource/YouTube, respectively. Both of the entities can be further looked up for their typing information expressed by DBpedia's taxonomical semantics (e.g. organization, broadcaster, company, etc. ontology classes).

The extracted information needs to be represented in a normalized, searchable format that enables adding unique knowledge assertions, coping with statements carrying duplicate knowledge and serving open- domain, free-text user's queries at the same time. For this, we employ Semantic Web technology. In particular, we propose an ontology capable of expressing both shallow and deep semantics behind natural language sentences. The extracted knowledge is serialized using Resource Description Framework (RDF) resulting in a directed labeled knowledge graph. Such representation further allows treating query execution as a graph traversal task. A constrained spreading activation algorithm is adapted to the proposed ontology schema with the main goal of emitting top-k graph nodes that best stand for user's needs behind the original query.

The rest of the paper is structured as follows. Section 2 provides an overview of related work in IR and IE fields. The proposed method for semantically enhanced IR is presented in detail in Sections 3-5. Section 6 discusses our experimental observations and lessons learned from method evaluation on a WikiQA dataset. Finally, we draw conclusions and further work directions in Section 7.

2. Related Work

With the emerging growth of Semantic Web technology, the way web IR has been seen is changing. Standard document text pre-processing steps like tokenization, stop word removal, stemming and lemmatization, etc. are getting complemented by more advanced information extraction methods. The introduction of common standards for semantic data and domain knowledge representation (RDF, RDFS, OWL), influenced a wide body of research towards meaning based IR, generally known under a term of semantic search [17, 8, 15, 13, 29].

Semantic search approaches can be classified on many criteria [17], such as conceptualization level of document content, query types for expressing information needs, methods for content ranking etc. Here, with the original research aim in mind, we focus on information extraction (IE) methods applied in semantic search proposals.

Kiryakov et al. [15] propose a semantically enhanced IE system called KIM that applies semantic document annotation in order to establish references between a document's named entities (NE) and ontology concepts and instances found in a pre-defined or automatically extracted knowledge base. Mapping NEs to a formal knowledge base instances allows taking advantage of implicit knowledge that can be derived by inferring different OWL/RDFS entailments. Moreover, Kiryakov et al. [15] use a traditional inverted index to store ordinary tokens along with special identifiers that link slices of text to particular entity URIs within the knowledge base. Such indexing scheme provides users the ability to search for documents by entity name restrictions disregarding the different aliases entities may have in the document text.

A semantic search system that extends generalized vector space model with taxonomic relationships is proposed in [8]. An inverted index is used to handle both textual and semantic content. This allows for the IR phase to be treated as combined query-document vector comparison task where words, entities and ontological classes all take part in vector representation. The ability to adapt classical inverted index data structures in such semantic IR proposals helps to avoid the necessity of dealing with natural structured/semi-structured language interfaces to semantic data [27, 14] as discussed below.

The work presented by Vileiniškis et al. [29] shows an example of ontology population- driven semantic search approach that targets the web corpus of Lithuanian news portals. Here, unlike the previously mentioned IE methods, a domain specific ontology T-box is filled with instance (A-box) data by extracting text fragments corresponding to particular ontology concept mentions within the processed text. Having a domain specific ontology filled with automatically extracted instance data implies the need for methods to ask queries against the derived knowledge base. As was mentioned before, formal SPARQL queries are barely an option from the perspective of a casual end user. To deal with this IR limitation, Vileiniškis et al. [29] employ a method of structured natural language question transformation to formal SPARQL queries. The approach follows the work by Sukys et. al. [27], where controlled natural language questions represented as SBVR models are transformed to SPARQL query models by M2M (model-to-model) transformation

techniques. Recent studies [3, 26] show that SPARQL query construction can be effectively approached from plain natural language questions as well.

In difference to all of the related work mentioned above, and most of the semantic IR approaches studied in literature, we focus on semantically-aware relation extraction between entities instead of standalone entity extraction from textual fragments. That is, we first seek to identify the relations expressed in a sentence via SRL and only then dig for deeper semantics. Similar approaches are used in research oriented towards event extraction, especially for abstractive news events summarization [23, 22, 18]. Multiple IE approaches [12, 9, 7] use shallow semantic parsing as basis for obtaining deeper semantics within labeled predicate arguments. However, there is no focus on the IR problem, i.e., utilization of the generated knowledge for serving user's information needs.

The closest work to ours is the approach to relational web search as proposed in [6]. An entity- relationship graph is constructed by applying lexical extraction patterns during the IE phase and spreading activation is employed as means for IR later on. However, no disambiguation techniques are used to normalize entities participating in various relations. In addition, we utilize external knowledge base (DBpedia) and semantic web technology that were not sophisticated enough/did not exist at the time of original research by the authors.

3. Semantically Enhanced IR

This section presents a conceptual IR model that combines SRL-driven information extraction method with a graph traversal algorithm for retrieval of the resulting predicate-argument structures. The model is depicted in Figure 1.

The general idea behind the model is to maintain a unique set of event-specific knowledge bits that are described by atomic tuples consisting of the predicate and its main arguments – subject (A0/A1) and object (A1/A2). We will refer to these knowledge bits as SRL triples. Triple extraction from unstructured text follows a pipeline of three main NLP components: semantic role labeler, named entity tagger and disambiguator (entity KB linker). As shown in Figure 1,

Figure 1

Conceptual model for SRL-based information retrieval

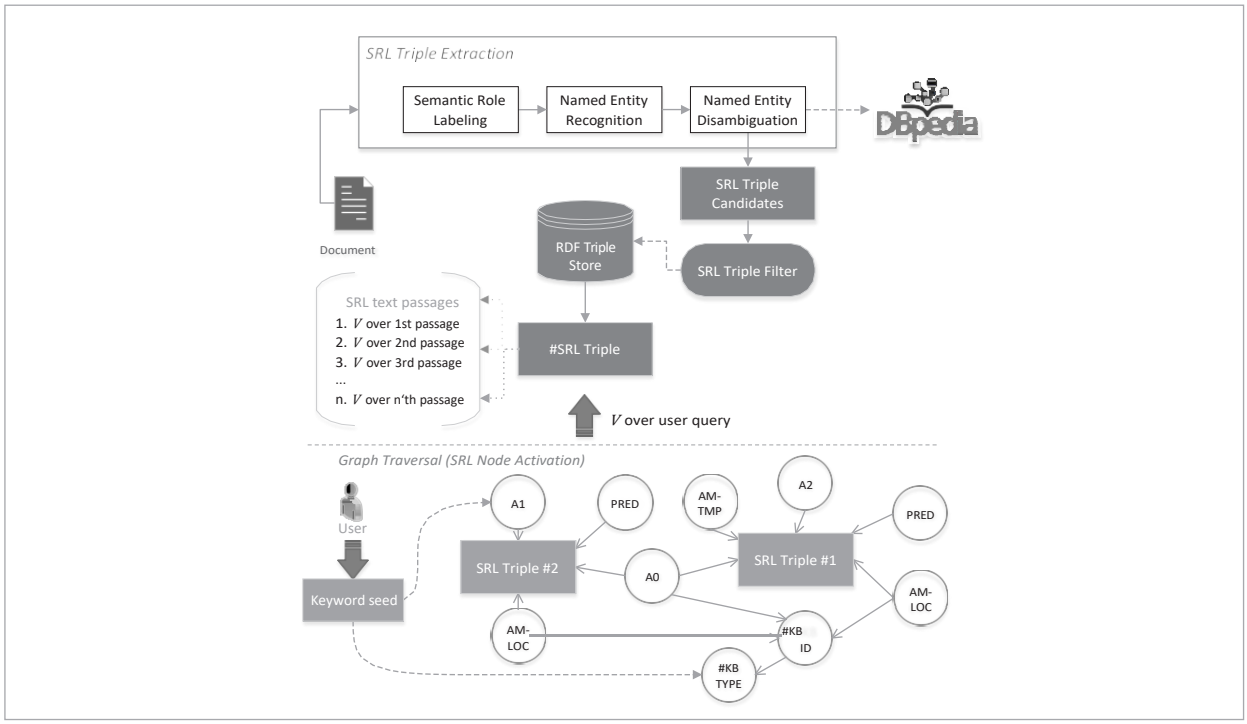
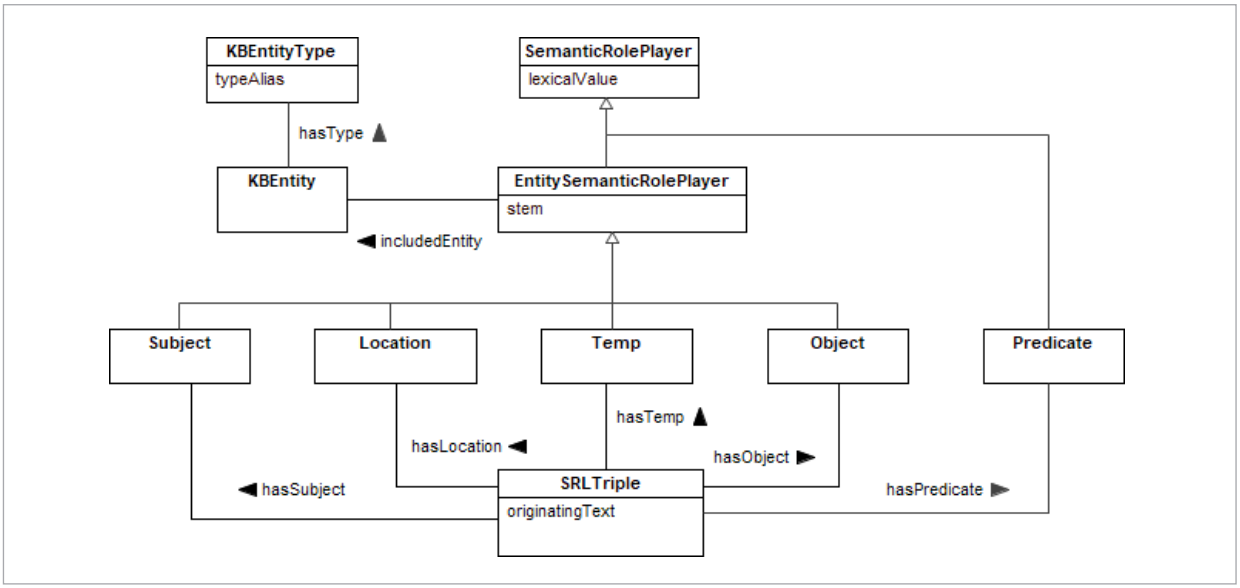


Figure 2
A schema of SRL triple ontology



each of the SRL triple candidates is first checked for uniqueness in the RDF store:

1 If the candidate tuple is found to be a duplicate, the extracted SRL text passage is weighted for inclusion in SRL triple's text passage list. The latter acts as a textual evidence for the knowledge carried by the SRL triple in question. Multiple quality measures can be employed to judge the quality of a text passage or the whole document it occurs in, e.g. semantic contextual similarity of the neighboring sentences or the amount of additional event characteristics present in secondary predicate arguments.

2 If the candidate tuple is found to be unique, a new assertion is made according to an ontology schema (see Figure 2) and the new text passage gets added to triple- bounded SRL text passage list.

This event-specific information extraction approach enables to maintain a collection of only unique event mentions and increases semantic storage requirements only when unseen, meaningful events occur.

4. SRL Triple Ontology

The SRL triple ontology is created for capturing and maintaining both shallow and deep semantics of a single SRL triple. These are the main classes and properties of the ontology:

SRLTriple – serves as the base of a new event- specific knowledge bit. Each valid SRL triple is characterized by a minimum of two object properties – "hasSubject" (denotes the subject part of the triple) and "hasObject" (denotes the object part of the triple). Other direct properties ("hasLocation", "hasTemp") are optional and do not determine the uniqueness of an atomic SRL fact.

Predicate – the predicate of the triple and its sense are maintained within this class. The set of available predicates is finite and should be constrained to the PropBank verb lexicon.

Subject – captures the subject-specific details of a SRL triple. In particular, datatype value "lexicalValue" stores the lexical alias of an extracted subject as-is. The disambiguated KB entity within subject position is retrieved by an object property "includedEntity". The relationship of a subject and its nested SRL triples is expressed by an object property "hasNested".

Object – captures the object-specific details of a SRL triple. The properties carry similar semantics as the ones described above for Subject class.

Temp – captures the optional AM-TMP argument of a SRL triple.

Location – captures the optional AM-LOC argument of a SRL triple. Unlike Temp, location-bearing event

Figure 3
RDF instance data in Turtle syntax

```
<http://semantika.srl/srl/srltriple/65c39081-7f00-4546-a689-fdcea52e4c16> a
<srl:SRLTriple>;
  <srl:hasObject> <http://semantika.srl/srl/object/cb3945e9-2024-4641-a027-
b44b372d9420>;
  <srl:hasPred> <http://semantika.srl/srl/predicate/buy.01>;
  <srl:hasSubject> <http://semantika.srl/srl/subject/0fc8b8d0-b048-40dd-a8ed-
b339a9f78d48>;
  <srl:originatingText> "In November 2006, YouTube was bought by Google for US$1.65
billion, and operates as a subsidiary of Google.".

<http://semantika.srl/srl/object/cb3945e9-2024-4641-a027-b44b372d9420> a <srl:Object>;
  <srl:includedEntity> <http://dbpedia.org/resource/YouTube>;
  <srl:lexicalValue> "YouTube";
  <srl:stem> "YouTub".

<http://semantika.srl/srl/subject/0fc8b8d0-b048-40dd-a8ed-b339a9f78d48> a
<srl:Subject>;
  <srl:includedEntity> <http://dbpedia.org/resource/Google>;
  <srl:lexicalValue> "by Google";
  <srl:stem> "Googl".
```

argument is worth to be disambiguated against public KB entries to determine the fine-grained type of the event location. This should enable more abstract "where" type queries at the IR phase.

KBEntity – models the entity that the subject, object or locative manner argument gets disambiguated against. Multiple knowledge bases could be used to disambiguate entities against. In our case, we rely on DBpedia.

KBEntityType – captures the notable type(s) of a particular KB entity. The way that the type could be extracted differs per knowledge base. In DBpedia's case, typing information can be retrieved following a simple rdf:type predicate.

EntitySemanticRolePlayer – parent class of the main and secondary SRL predicate argument classes where NEs are expected to take participation.

SemanticRolePlayer – parent class of all of the SRL argument classes, predicate included.

It is worth noting that this kind of ontology schema is not aimed at capturing domain- specific event knowledge. It is focused towards more "open" information extraction paradigm where event semantics are determined by shallow linguistic features of SRL structures. The deep semantics are limited to fine-grained ontological typing of main SRL arguments.

The advantage of having SRL Triples serialized in RDF form is the ability to check for existence of duplicate event-knowledge using SPARQL queries (see Figure 4).

Figure 4
SPARQL query for knowledge duplication checking

```
ASK WHERE {
  ?SRL a :SRLTriple.
  ?SRL :has_pred <PRED#SENSE>.
  ?SRL :has_subject ?sub.
  ?sub :included_entity <A0#KB_URI>.
  ?SRL :has_object ?obj.
  ?obj :included_entity <A1#KB_URI>.
}
```

The example SPARQL ASK query above would result in a Boolean answer once executed over the instantiated ontology – *true* would suggest that the atomic SRL triple already exists, *false* – that the SRL triple is missing and should be asserted as a new event-knowledge bit.

Since there can be multiple entities detected in either subject or object arguments of a predicate, the highlighted triple patterns are generated for each occurrence of a NE. That is, having sets E_{sub} (named entities

in subject position) and E_{obj} (named entities in object position):

$for\ e_{sub} \in E_{sub} \rightarrow ?sub\ :included\_entity\ e_{sub}$

$for\ e_{obj} \in E_{obj} \rightarrow ?sub\ :included\_entity\ e_{obj.}$

Figure 3 provides a sample illustration of how the ontology schema gets instantiated with RDF triples, upon extracting knowledge from a sample sentence (D_1).

5. Interpreting User's Information Needs

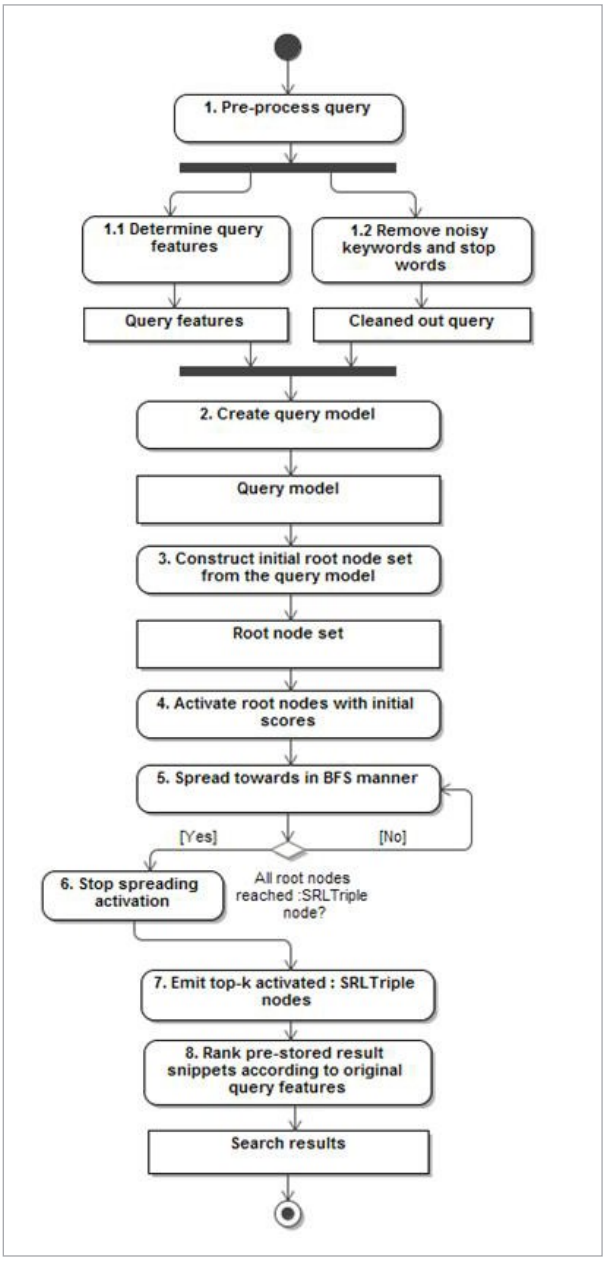
The proposed IR model allows free word order, unrestricted user queries, which is a common strategy among current major web search engines. The goal here is to attempt to derive a SRL triple(s) that the user is most likely interested in from a set of input keywords. The general flow of such IR algorithm is depicted in Figure 5.

At the core of the IR model lays spreading activation – a graph traversal algorithm that tries to mimic semantic processing of human brain as studied in cognitive psychology. Its applicability for navigating any associative network is employed to traverse the constructed RDF knowledge graph in a semantically controlled manner – travelling is enforced to be routed through semantically associated nodes and the importance of the final target node judged by summed semantic features met on the pathway.

Let us explain the main steps of our proposed algorithm:

1. First, the query is preprocessed by:
   a. Stop words and other noisy keywords are removed from the query.
   b. Remaining query terms are tagged by a NE tagger and disambiguated against a knowledge base (DBpedia in our case).
2. Both the original plain query keywords and the NEs get serialized into a common model – a JSON data structure that represents semantic query features (URIs of disambiguated entities) along with keyword-like query term stems.
3. An initial root node set is constructed of all of the RDF graph nodes (URIs) that have query keywords in their lexical values. In addition to those, URIs of disambiguated NEs are also included in the set. To ease the construction of the initial node set, during IE we create and maintain an inverted keyword → URI index, where keywords are the tokenized lexical values of predicate arguments and the predicate itself. This way we get rid of meaningless tokens

Figure 5
Activity flow of spreading activation-based IR algorithm



that do not play any role in the sentence, effectively reducing unnecessary noise in search results.

4 Scores are assigned to nodes in the initial set. We assign a default value of 1 to every node that has only a keyword match and a value of 3 to a node that either stands as a DBpedia entity or matches a lexical value of a DBpedia concept. This way, we prioritize energy spreading from nodes that hold more semantic power in contrast to plain keyword ones. Keyword frequency is taken into account as well – an exact amount of query keyword → URI index matches gets added to the initial node score to boost its initial energy and importance query-wise.

5 The resulting sub-graph then gets traversed by a graph traversal algorithm - a modified version of spreading activation to find which of the SRL triple nodes gets activated the most, i.e. which of the SRL triples is most likely behind user's information needs. Unlike traditional spreading activation constraints (travel distance, activation strength threshold etc.), in our case spreading stops once a node (ontology class instance) of the type "srl:SRLTriple" is met on the pathway – this signals of a semantically closest event- knowledge bit tied to the given query. The scores of all the initial nodes that reached the target are summed and presented as final SRL node activation score.

6 Top-k SRL triple nodes that reach the highest activation scores are picked for result presentation.

7 As one single "srl:SRLTriple" node can have multiple text passages as proofs of the captured knowledge, re-ranking techniques can be applied to sort those by relevancy. However, the quality aspects of multiple text passages are not in the scope of this paper, therefore, such re-ranking will not be considered in the remaining sections. Occurrences of multiple textual proofs are considered to be equal.

The spreading activation logic is based on Breadth First Search (BFS) graph traversal principle (see Algorithm 1).

Going back to the running sentence example (D_1), Figure 6 shows how the extracted SRL triple knowledge bit appears in the RDF graph.

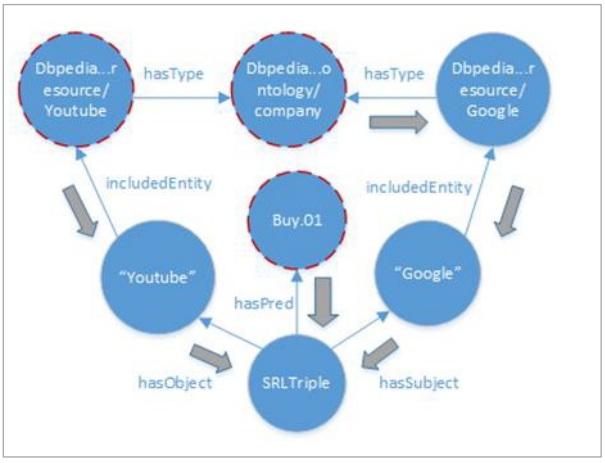
Given a sample query "Which company bought Youtube?", the nodes with a dashed border would get included in the initial root node set as a starting point for graph traversal. As can be seen from the graph,

Algorithm 1 Constrained spreading activation

INPUT: N - intial root node set
OUTPUT: R - result set of top-k SRLTriple nodes

```
1: srlTripleNodeReached ← FALSE
2: NeighborNodes ← ∅
3: procedure SpreadngActivation(N)
4:     for each node n ∈ N do
5:         SetInitialActivationScore(n)
6:         NeighborNodes ←SpreadInBFSManner(n)
7:         for each node neighbour ∈ NeighbourNodes do
8:             while srlTripleNodeReached ≠ TRUE do
9:                 if neighbour is of type :SRLTriple then
10:                    srlTripleNodeReached ← TRUE
11:                    add neighbour to R
12:                else
13:                    continue SpreadInBFSManner(neighbour)
14:                end if
15:            end while
16:            srlTripleNodeReached ← FALSE
17:        end for
18:        NeighborNodes ← ∅
19:        sort R by activation score
20:    end for
21: end procedure
```

Figure 6
A visual graph representation of extracted factfrom sentence D_1



the maximum number of edges to be traveled through by spreading activation algorithm is 3, given that an ontology concept is among the initial nodes. At the moment, semantics of the edges themselves is not considered during the graph walk, since no further analysis is being made on the free text input query, apart from NE tagging and disambiguation. Therefore, the final scores are judged only by the graph node semantics. In this case, the total activation value for the target SRLTriple node would be 3+3+1=7.

Serving user's information needs via spreading activation on top of the extracted knowledge graph allows accepting free word order queries as an input while still respecting semantic predicate-argument relations of nodes during query execution. That is, the target answer-bearing SRLTriple node can only be reached through other graph nodes if there is a linguistic dependency between them. This is an advancement compared to usual full-text search approaches where term-document matching does not take into account the underlying shallow linguistic features of indexed text, resulting in knowledge-agnostic document representation and subsequently narrow querying abilities with low precision.

6. Experimental Evaluation

An evaluation of the proposed semantically enhanced IR method was conducted by firstly applying our SRL Triple Extraction IE component on WikiQA [30] full dataset, and secondly by using corresponding query set to see how spreading activation algorithm behaves on top of the extracted RDF knowledge graph.

The choice of evaluation method when it comes to semantic search approaches (and ours in particular) is not a straightforward task. First, the scope of tackled research problems differs. Second, it is the ambiguity and wide interpretation of the term "semantic". Most of the evaluation initiatives (SemSearch [16], QALD [25]) focus on already existing structured RDF knowledge bases where the main goal is to provide objects (URIs) as answers to the queries. In our case, we start from unstructured text to create a specific RDF knowledge base with a graph structure aimed at serving free-text queries during IR phase in a semantic manner. By the term 'semantic', we mean having an ability to interpret implicit information needs not necessarily explicitly available in the target corpora. Hence, both the IE and IR phases are to be evaluated jointly.

Question answering (QA) evaluation datasets were chosen to be the best fit to assess the semantic search approach presented in this paper. In particular, WikiQA provides both question and sentence pairs in an open-domain space allowing to first, apply IE on the unstructured data, and later, utilize the structured RDF knowledge graph for the IR task. As many of the questions in the dataset tend to vary from event-seeking, entity- oriented ones to instance-class relationships, they perfectly fit by our definition of semantic search.

While QA systems are usually expected to emit text fragments as precise answers to questions given a corresponding sentence/paragraph, our solution differs as it finds top-k SRLTriple nodes behind expressed user's information needs in the whole RDF knowledge graph and provides full sentences as textual proofs of the aforementioned facts. This way, we still maintain the context of the query answer which is very similar to how major internet search engines still work nowadays.

As opposed to typical QA system evaluation methods, we chose to evaluate triple extraction and information retrieval effectiveness instead. That is, the behavior of the proposed solution was analyzed by digging deeper into different algorithm characteristics and observing the influence it has on the end search results. Our goal here is to build a baseline ourselves which could be used in future research.

Lastly, we chose to compare our solution to classical inverted text indexing approach that utilizes TF-IDF weighting scheme and Vector Space Model for IR task.

6.1. Dataset Pre-processing

WikiQA consists of many different variety question-sentence pairs and not all of them are suitable for evaluation in our case. Hence, out of 1473 sentences that have at least one correct answer to a corresponding question, 1025 were filtered as target for annotation. Filtering was carried out in order to end up with a subset of sentences that have at least one NE mention. This helps to avoid unnecessary SRL annotations for sentences that are not targeted by our current research, where existence of a NE is a crucial criterion when forming valid SRL triples.

6.2. SRL Triple Extraction

A full text annotation pipeline has been implemented as shown at the top of Figure 1. The pipeline consists of three main NLP components:

- SRL annotator – produces text annotations in predicate-argument structure [24].
- NER annotator – marks named entities (NEs) in text to be used by NED component [10].
- NED annotator – disambiguates NEs against a knowledge base. DBpedia is being used in our case [28].

Unique SRL triple knowledge bit assertion rules have been implemented following the annotations produced by the above pipeline. SRL triples were asserted as RDF statements into an RDF data store.

A variant of constrained spreading activation algorithm has been implemented to operate on top of RDF data structures for direct IR purposes. For query pre-processing, we used the same NER and NED components as mentioned above.

6.3. Information Extraction Results

The annotation of the filtered WikiQA corpus finished with a total of 102 unique SRL triples asserted from 84 sentences (79 documents). Such low annotation recall value (8.2%) can be explained by the following:

1. SRL triple knowledge bit assertion rules are very strict, requiring at least one disambiguated and linked NE within both main predicate arguments A0/A1 or A1/A2. (<NE, PRED, NE>)
2. The SRL annotator used does not perform very well on open-domain texts, since its models are trained on domain- specific data. Hence, flaws are very common in both argument identification and role assignment.
3. The SRL annotator skips predicate "be.01" making it a drawback when dealing with factoid-like questions in WikiQA dataset as the required triples do not get asserted in the knowledge base during information extraction.

We believe that switching SRL component in the IE pipeline to a more sophisticated one that is capable of better handling open- domain texts should improve annotation recall quite significantly.

6.4. Information Retrieval Results

Having the SRL Triple ontology populated with instance data, we gathered a list of queries in WikiQA corpus that correspond to successfully annotated sentences. That is, we eliminated queries that we are for sure not capable of answering since IE pipeline did not produce valid SRL triple assertions for specific query-document pairs. This left us with 91 queries, out of which our system managed to successfully emit correct answer nodes for 62. Table 1 shows resulting query execution characteristics on a fragment of the successfully processed queries. To analyze the behavior of spreading activation algorithm, we compare the activation metrics for answer-bearing RDF graph node in context of all of the nodes activated during the run.

The spreading activation algorithm-based IR flow tends to reach expected graph nodes 68.1% of the time; however, their sum activation values do not seem to be necessarily the highest out of all the nodes reached during query execution. In particular, expected nodes got the highest scores for 64.5% of the queries. Failure for 35.5% ones is usually the case when initial root node set is faulty because of relaxed full-text-like matching of query terms to graph node lexical values (see Figure 5, step #3). Even though such keyword index lookup strategy acts like a backup in cases when there is no semantic match for the query in the knowledge base (no NE, no entity type among the keywords), the side effect of introducing irrelevant noise in the search results is still

apparent. This mainly has to do with long-tail argument lexical values (especially the ones of object role) suggesting a need for noun-phrase mining instead of full indexing of the entire role text.

Pure keyword-like matching to DBpedia entity type (ontology class) URIs can also cause topic drift from initial query needs. E.g., For a query (Q1776): "What year did South Africa become a team in rugby?", the query keyword "team" matches URIs like:

http://dbpedia.org/ontology/SportsTeam, http://dbpedia.org/ontology/BaseballTeam, http://dbpedia.org/ontology/BasketballTeam.

That eventually leads the spreading activation graph traversing algorithm to reach nodes with instance mentions of the above class instances, resulting in incorrect target SRL triple nodes activated more than expected ones.

The remaining 31.9% of queries were not served at all. The root cause analysis has shown that most of these queries are suffering from currently non- existent semantic keyword expansion for predicates and inefficient ontological NE typing because of DBpedia's pure taxonomical concept semantics. That is, ontology classes and in particular, their labels, are highly unlikely to be often encountered in user queries.

6.5. Comparison with TF-IDF

Comparing the performance of our system to an existing baseline is not trivial as we could not find analogous research carried out by other authors. QA

Table 1
Spreading activation statistics of successful execution of 30 random WikiQA queries

QueryID	# of Activations	Activation Score	Max # of Activations	Spreading started from concept	Spreading started from NE	# of graph nodes emmited	Max Activation Score
Q11	1	2	1	FALSE	FALSE	3	2
Q26	2	5	2	FALSE	FALSE	7	5
Q75	1	4	1	FALSE	FALSE	2	4
Q181	1	2	2	FALSE	FALSE	8	4
Q195	2	5	2	FALSE	FALSE	2	5
Q304	3	6	3	FALSE	FALSE	5	6
Q398	2	7	2	FALSE	FALSE	11	8
Q492	1	2	1	FALSE	FALSE	8	2
Q557	1	2	2	FALSE	FALSE	25	8
Q619	1	2	1	FALSE	FALSE	3	2
Q622	2	6	2	FALSE	TRUE	14	6
Q679	1	2	1	FALSE	FALSE	1	2
Q682	2	4	2	FALSE	FALSE	17	4
Q1014	1	2	1	FALSE	FALSE	1	2
Q1046	1	2	2	FALSE	FALSE	10	10
Q1069	1	3	1	FALSE	FALSE	2	3
Q1075	3	10	3	TRUE	FALSE	58	12
Q1257	2	5	2	FALSE	FALSE	9	5
Q1262	2	6	2	FALSE	TRUE	7	6
Q1284	1	2	1	FALSE	FALSE	2	2
Q1519	2	6	2	FALSE	FALSE	11	6
Q2129	1	4	2	TRUE	FALSE	17	8
Q2221	1	4	1	FALSE	FALSE	6	4
Q2293	2	5	2	FALSE	FALSE	23	5
Q2635	1	2	2	FALSE	FALSE	4	4
Q2675	2	5	4	FALSE	FALSE	84	12
Q2797	3	13	3	FALSE	TRUE	23	13
Q2978	1	2	1	FALSE	FALSE	6	2
Q2999	2	9	2	FALSE	FALSE	23	9
Q3010	2	4	2	FALSE	FALSE	1	4
Q3027	1	3	1	FALSE	FALSE	1	3

systems participating in WikiQA challenge are also aimed at different goals of providing precise answers to questions rather than emitting best matching documents/sentences as we do.

Therefore, we chose to evaluate our system against the classical inverted full text index and TF-IDF weighting scheme by utilizing Lucene framework [1]. As presented in Section 6.1, we indexed the filtered down dataset of 1025 sentences under *StandardAnalyzer* and *ClassicSimilarity* settings. The very same 91 queries were taken from Section 6.4 in order to end up with the same query set for both compared solutions. Since text tokenization and stop word removal performed with *StandardAnalyzer* setting in Lucene cannot be considered as comprehensive IE tasks (more like text pre-processing), in Table 2 we report the performance results of IR only. The query is treated as served, if the system manages to emit an answer-bearing sentence among all of the output results. For the query to be correctly served, the answer is expected to be either a node with the highest activation value (our system) or a sentence scored the highest by a Vector Space Model (VSM) score in Lucene.

Table 2
IR comparison results between SRL and Lucene systems

Name	Query Set	Served Queries	Correctly Served Queries	Recall	Precision
SRL	91	62	40	44%	64.5%
Lucene	91	85	41	45%	48.2%

As expected, Lucene manages to serve significantly more queries (85 out of 91) since for that a Boolean match between a single query and document term is enough. However, such plain keyword matching approach falls short to serve the queries effectively. In particular, even suffering from limited query term expansion abilities in its current form, our system reaches better precision value by 16.3%. This is a result of leveraging predicate- argument structures as atomic knowledge bits for data indexing and performing query term matching in linguistically-controlled manner, instead of solely relying on semantically- agnostic term existence/popularity comparison in TF-IDF and VSM techniques.

7. Conclusions

This paper introduced an approach to semantically enhanced IR. We proposed a method that combines SRL-based IE and spreading activation-driven IR to enable both capturing and querying unique knowledge from unstructured text. An experimental evaluation has shown that our approach already outweighs the classical Vector Space Model and TF-IDF method, thus is suitable to tackle semantic IR from the conceptual perspective. Fine-graining indexing and querying scope to predicate-argument structures improves event-seeking query precision significantly by eliminating constituents of a sentence that do not take part in any semantic role, thus are unimportant event-wise.

We identified a number of enhancements that should be considered for future work. First, the quality of IE pipeline, namely, consisting of SRL, NE and NED components should be reviewed and judged for applicability in an open-domain setting. Having the components trained on domain-specific data has a significant negative impact on annotation phase recall values, hence prohibiting more sophisticated analysis due to lack of annotated data. Second, <NP, PRED, NP> (NP – Noun Phrase) knowledge extraction rule might be a better fit for open-domain texts as the <NE, PRED, NE> one seems to be a bit too strict, at least for WikiQA corpora, where a named entity in both subject and object positions of a predicate is not so common. However, such relaxation would introduce additional disambiguation challenges for the predicate arguments. Third, spreading activation algorithm scoring mechanism could be improved to better cope with noise caused by faulty plain keyword-based initial root node selection. Finally, we seek to extend SRL argument analysis beyond ontological typing. In particular, nested SRL structures could reveal more implicit knowledge once morphosemantic heuristics are employed on top of them.

References

1. Apache Lucene, https://lucene.apache.org/.
2. Baker, C. F., Fillmore, C. J., Lowe, J. B. The Berkeley Framenet Project. In Proceedings of the 17th International Conference on Computational Linguistics-Volume 1, Association for Computational Linguistics, 1998, 86-90. https://doi.org/10.3115/980451.980860
3. Bakhshi, M., Nematbakhsh, M., Mohsenzadeh, M. and Rahmani, A. M. Data-Driven Construction of SPARQL Queries by Approximate Question Graph Alignment in Question Answering Over Knowledge Graphs. Expert Systems with Applications, 2020, p.113205. https://doi.org/10.1016/j.eswa.2020.113205
4. Björkelund, A., Hafdell, L., Nugues, P. Multilingual Semantic Role Labeling. In Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task, Association for Computational Linguistics, 2009, 43-48. https://doi.org/10.3115/1596409.1596416
5. Cai, R., Lapata, M. Syntax-aware Semantic Role Labeling Without Parsing. Transactions of the Association for Computational Linguistics, 2019, 7, 343-356. https://doi.org/10.1162/tacl_a_00272
6. Cafarella, M. J., Banko, M., Etzioni, O. Relational Web Search. In WWW Conference, 2006.
7. Christensen, J., Soderland, S., Etzioni, O. An Analysis of Open Information Extraction Based on Semantic Role Labeling, In Proceedings of the Sixth International Conference on Knowledge Capture, ACM, 2011, 113-120.
8. Corcoglioniti, F., Dragoni, M., Rospocher, M. Aprosio, A. P. Knowledge Extraction for Information Retrieval. In European Semantic Web Conference, Springer, Cham, 2016, 317- 333. https://doi.org/10.1007/978-3-319-34129-3_20
9. Exner, P., Nugues, P. Using Semantic Role Labeling to Extract Events from Wikipedia. In DeRiVE@ ISWC, 2011, 38-47.
10. Finkel, J. R., Grenager, T., Manning, C. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, ACM, 2005, 363-370. https://doi.org/10.3115/1219840.1219885
11. Gaizauskas, R. J., Robertson, A. M. Coupling Information Retrieval and Information Extraction: A New Text Technology for Gathering Information from the Web. In RIAO, 1997, 356- 370.
12. Haarmann, B., Sikorski, L., Schade, U. Text Analysis beyond Keyword Spotting. In Proceedings of the Military Communications & Information Systems Conference (MCC), 2011.
13. Jiang, Y., Yang, M. Semantic Search Exploiting Formal Concept Analysis, Rough Sets, and Wikipedia. International Journal on Semantic Web and Information Systems (IJSWIS), 2018, 14(3), 99-119. https://doi.org/10.4018/IJSWIS.2018070105
14. Kaufmann, E., Bernstein, A. How Useful Are Natural Language Interfaces to the semantic Web for Casual End-users? In The Semantic Web, Springer Berlin Heidelberg, 2007, 281-294. https://doi.org/10.1007/978-3-540-76298-0_21
15. Kiryakov, A., Popov, B., Terziev, I., Manov, D., Ognyanoff, D. Semantic Annotation, Indexing, and Retrieval, Web Semantics: Science, Services and Agents on the World Wide Web, 2004, 2(1), 49-79. https://doi.org/10.1016/j.websem.2004.07.005
16. Lopez, V., Unger, C., Cimiano, P., Motta, E. Evaluating Question Answering Over Linked Data. Web Semantics: Science, Services and Agents on the World Wide Web, 2013, 21, 3-13. https://doi.org/10.1016/j.websem.2013.05.006
17. Mangold, C. A Survey and Classification of Semantic Search Approaches. International Journal of Metadata, Semantics and Ontologies, 2007, 2(1), 23-34. https://doi.org/10.1504/IJMSO.2007.015073
18. Mohamed, M., Oussalah, M. SRL-ESA- TextSum: A Text Summarization Approach Based on Semantic Role Labeling and Explicit Semantic Analysis. Information Processing & Management, 2019, 56(4), 1356-1372. https://doi.org/10.1016/j.ipm.2019.04.003
19. Ofoghi, B., Yearwood, J., Ma, L. The Impact of Semantic Class Identification and Semantic Role Labeling on Natural Language Answer Extraction. In Advances in Information Retrieval, Springer Berlin Heidelberg, 2008, 430-437.
20. Palmer, M., Gildea, D., Kingsbury, P. The Proposition Bank: An Annotated Corpus of Semantic Roles. Computational Linguistics, 2005, 31(1), 71-106. https://doi.org/10.1162/0891201053630264
21. Pizzato, L. A., Mollá, D. Indexing on Semantic Roles for Question Answering. In Coling 2008: Proceedings of the 2nd workshop on Information Retrieval for Question

Answering, Association for Computational Linguistics, 2008, 74-81. https://doi.org/10.3115/1641451.1641461

22. Prasojo, R.E., Kacimi, M., Nutt, W. Modeling and Summarizing News Events Using Semantic Triples. In European Semantic Web Conference, Springer, Cham, 2018, 512-527. https://doi.org/10.1007/978-3-319-93417-4_33
23. Rospocher, M., van Erp, M., Vossen, P., Fokkens, A., Aldabe, I., Rigau, G., Soroa, A., Ploeger, T., Bogaard, T. Building Event-Centric Knowledge Graphs from News. Journal of Web Semantics, 2016, 37, 132- 151. https://doi.org/10.1016/j.websem.2015.12.004
24. Roth, M., Woodsend, K. Composition of Word Representations Improves Semantic Role Labelling. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, 407-413. https://doi.org/10.3115/v1/D14-1045
25. Semantic Search Challenge, 4th International Semantic Search Workshop March 29, 2011 (http://km.aifb.kit.edu/ws/semsearch11/)
26. Song, S., Huang, W., Sun, Y. Semantic Query Graph Based SPARQL Generation from Natural Language Questions. Cluster Computing, 2019, 22(1), 847-858. https://doi.org/10.1007/s10586-017-1332-3
27. Sukys, A., Nemuraite, L., Paradauskas, B. Representing and Transforming SBVR Question Patterns into SPARQL. In Information and Software Technologies, Springer Berlin Heidelberg, 2012, 436-451. https://doi.org/10.1007/978-3-642-33308-8_36
28. Usbeck, R., Ngomo, A. C. N., Röder, M., Gerber, D., Coelho, S. A., Auer, S., Both, A. AGDISTIS-graph-based Disambiguation of Named Entities Using Linked Data. In International Semantic Web Conference, Springer, Cham, 2014, 457-471. https://doi.org/10.1007/978-3-319-11964-9_29
29. Vileiniškis, T., Šukys, A., Butkienė, R. An Approach for Semantic Search Over Lithuanian News Website Corpus, IC3K 2015. In Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, 2015, Volume 1: KDIR (IC3K 2015), 57-66. https://doi.org/10.5220/0005596800570066
30. Yang, Y., Yih, W. T., Meek, C. Wikiqa: A Challenge Dataset for Open-domain Question Answering. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, 2013-2018. https://doi.org/10.18653/v1/D15-1237