An Approach for Semantic Search over Lithuanian News Website Corpus

Tomas Vileiniškis, Algirdas Šukys and Rita Butkienė
Department of Information Systems, Kaunas University of Technology, Studentu st. 50-309, Kaunas, Lithuania

Keywords: Semantic Search, SBVR, SPARQL, Information Retrieval, Ontology, Semantic Annotation, Lithuanian Language.

Abstract: The continuous growth of unstructured textual information on the web implies the need for novel, semantically aware content processing and information retrieval (IR) methods. Following the evolution and wide adoption of Semantic Web technology, a number of approaches to overcome the limitations of traditional keyword-based search techniques have been proposed. However, most of the research concentrates on English and other well-known, linguistic resource-rich languages. Hence, this paper presents an attempt to semantic search over domain-specific Lithuanian web documents. We introduce an ontology-based semantic search framework capable of answering structured natural Lithuanian language questions and discuss its language-dependent design decisions. The findings from a recent case study showed that our proposed framework can be applied to approach meaning-based IR with significant results, even when the underlying language is morphologically rich and has limited linguistic resources.

1 INTRODUCTION

In the context of traditional Web search, Information Retrieval (IR) has been known as a task of retrieving documents relevant to user information needs, typically expressed by some form of a query. A general IR model consists of three major building blocks: representation of a user query, document content description and a retrieval function. Early work in IR field highly focused on keyword-based models, such as exact-match Boolean and statistical Vector Space Model (Salton et al., 1975). The obvious shortcoming of these models is the lack of conceptualization both at the query and document representation level, which eventually results in poor precision and recall rates. A number of approaches such as query expansion (Carpineto and Romano, 2012) and word sense disambiguation (Stokoe et al., 2003) have been proposed to manage synonymy and polysemy in order to overcome the limitations of prior models to some extent.

However, with the emerging growth of Semantic Web technology, the way Web information retrieval has been seen is changing. The introduction of common standards for semantic data and domain knowledge representation (Resource Description Framework (RDF), RDF Schema (RDFS), Web Ontology Language (OWL2)) followed by a dedicated RDF query language (SPARQL) influenced a wide body of research (Mangold, 2007) towards meaning-based IR, which we will refer to as *semantic search* throughout the paper. Standard document text preprocessing steps used in classical IR models (tokenization, stop word removal, stemming etc.) are getting complemented by more advanced Information Extraction (IE) methods such as semantic annotation and ontology population. We believe that application of IE methods for content processing builds the foundation for efficient semantic search.

In general, IE is known as an activity of automatically extracting structured information from unstructured information source. The main challenge here is the complexity and ambiguity of natural language, hence making IE hardly dependent on advances in Natural Language Processing (NLP) techniques. While state-of-the-art in IE related NLP research for well-known languages (e.g. English) has already reached levels of successful practical application on a massive scale (e.g. IBM's Watson project) (Ferrucci, 2010), less popular and resource-poor languages such as Lithuanian, remain an open NLP research field.

The nature of Lithuanian language imposes many NLP-related challenges. First of all, it is highly inflected, which means that a single word root can lead to hundreds of different word forms, each of them expressing a distinct grammatical category. For

Vileiniškis, T., Šukys, A. and Butkienė, R..
An Approach for Semantic Search over Lithuanian News Website Corpus.
In *Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2015) - Volume 1: KDIR*, pages 57-66
ISBN: 978-989-758-158-8

example, a nominative singular noun *asmuo* (person) alone has multiple other grammatical cases reflected by alternating suffixes: *asmens* (genitive), *asmeniui* (dative), *asmenį* (accusative), *asmeniu* (instrumental), *asmenyje* (locative) etc. Such declension of nouns and adjectives plays a major role when determining grammatical function of a word in a sentence. Moreover, Lithuanian language doesn't have a strict word order. A single sentence can be expressed in multiple ways by switching word positions without losing the initial meaning. Therefore, reusing standard syntactic parsing approaches, applicable for strict word order languages (e.g. English) becomes complicated (Šveikauskienė and Telksnys, 2014). These and many other language-specific features require special attention when developing sophisticated IE methods dedicated for Lithuanian or any other morphologically rich languages (e.g., Slavic) in general.

In this paper we present a combined attempt to semantic content processing and search over Lithuanian web texts. A semantic search framework for the task is proposed. We introduce an ontology population-driven IE approach tightly coupled with a model-to-model (M2M) transformation-based IR model. We show how such tight-coupling enables us to serve natural structured language queries over domain-specific data represented in the form of ontology. We then evaluate applicability of our framework by performing a case study over Lithuanian news website corpus, focusing on political and economic domains. To the best of our knowledge, this is the first public attempt to Lithuanian text processing at the level of ontological semantics.

The rest of the paper is structured as follows. Section 2 gives a brief overview of related work in semantic search area and provides the state-of-the-art of NLP research for Lithuanian language. Section 3 presents the architecture of our semantic search framework with emphasis on capturing and maintaining domain-specific semantics throughout the search process. The experimental observations and lessons learned from the case study are presented in Section 4. Finally, we draw conclusions and discuss our future research plans in Section 5.

2 RELATED WORK

The evolution of Semantic Web technology has made a significant impact on meaning-based IR methods over the last decade. In particular, the introduction of W3C's OWL2, RDFS, RDF and SPARQL to conceptualize, represent and query domain specific knowledge led to an upsurge of research in the field.

(Kiryakov et al., 2004) proposed KIM - a framework for semantic annotation and retrieval. The main idea behind KIM is the semantic typing of named entities (NE) by linking them to pre-populated knowledge base entries and/or appropriate domain-ontology classes. (Fernández et al., 2011) introduced an approach for semantically enhanced IR by adapting the classical vector space model (Castells et al., 2007). The IE task used to conceptualize document content is similar to the one proposed by (Kiryakov et al., 2004). In addition, (Fernández et al., 2011) use an ontology-based Question Answering (QA) system to interpret the intent behind user queries. This is achieved by deriving linguistic triples from a natural language question and then looking up for answer-bearing ontology concepts by syntactic triple similarity matches (Lopez et al., 2009). Our approach to capturing user query intents differs substantially: we aim at obtaining a formal SPARQL query model from a structured natural language question (see Section 3.2).

Knowledge bases like Freebase or DBpedia have been recently used to tackle the problem of open-domain QA (Yao and Van Durme, 2014; Shekarpour et al., 2015). While their main goal is to retrieve answers to factoid-like questions over structured world's knowledge, our framework is primarily aimed at mining and searching domain-specific texts in order to satisfy event-oriented information needs.

All of the above mentioned approaches target semantic search only from an English language perspective, thus they build upon sophisticated NLP methods that are well known and properly researched. However, Lithuanian NLP research progresses in little steps. Perhaps one of the most significant achievements is the early work by (Zinkevičius, 2000) who created the first Lithuanian lemmatizer and part-of-speech (POS) tagger called *Lemuoklis*. The syntax of Lithuanian language has been extensively analyzed by (Šveikauskienė, 2005) (Šveikauskienė and Telksnys, 2014). A recent approach to statistical dependency parsing (Kapociute-Dzikiene et al., 2013) showed the importance of morphological features (especially grammatical case) for the accuracy of results. However, the lack of syntactically annotated data suggests that rule-based parsing is a better choice.

The only publically available case study of NLP-based content processing is presented in (Krilavičius

et al., 2012), where authors apply named entity recognition (NER) among other standard text pre-processing steps to annotate and analyse Lithuanian news media websites.

3 SEMANTIC SEARCH FRAMEWORK

The architecture of our proposed semantic search framework is depicted in Figure 1. As was noted in Section 1 of the paper, the framework consists of two major tightly coupled parts: the first one performs information extraction (IE) related tasks and the second one handles information retrieval (IR) related activities. For a detailed explanation please refer to Sections 3.1 and 3.2 respectively.

The IE module is dedicated for document text preprocessing and annotation by linguistic components in the NLP pipeline. IE module aims at capturing and conceptualizing entities and the events they participate in. We will emphasize the principles behind the semantic annotation component in subsequent sections. In order to avoid possible confusion about terminology, a note on the use of the terms "*semantic annotation*" and "*ontology population*" should be given (Amardeilh, 2008). Our text processing efforts concentrate on *ontology population*, i.e., adding instance data (*A-Box*) to a predefined ontology (*T-Box*). In addition, we perform *semantic annotation*, i.e., we link slices of text to their formal ontological representation bits (*A-box*) created in the *ontology population* step. In this aspect, our approach slightly differs from previous works discussed in Section 2. The reason for this is two-fold. First, there is no semantic knowledge base that would have sufficient coverage of domain specific entities and relations commonly mentioned in Lithuanian media. The construction of such resource would require a significant amount of manual labor. Although, the existence of multilingual lexical knowledge bases (e.g. BabelNet) (Navigli et al., 2012) is well-known, the entries for entities of a *local importance* (Lithuanian politics, organizations etc.) are rare to be found. Secondly, IR model behind our framework is based on formal SPARQL query execution, thus we expect for all the relevant domain knowledge acquired during text processing to be present in RDF form at query time.

The IR module behind the framework is highly based on SBVR (Semantics of Business Vocabulary and Business Rules) standard. SBVR is the OMG created metamodel and specification that defines vocabulary and rules for describing business semantics – business concepts, business facts, and business rules using some kind of Controlled Natural Language (OMG, 2008). SBVR enables to create formal specifications understandable for business people and also interpretable by software tools. This is achieved by the usage of structured natural language for representing meaning as formal logic structures – semantic formulations. SBVR metamodel is based on principle of separating meaning of business concepts and business restrictions from their representation. A number of transformations of SBVR specifications to various software models have been created: Web services (Goedertier and Vanthienen, 2008), BPMN (Bodenstaff et al., 2008), OWL2 (Karpovič et al., 2014), etc. We employ specific SBVR metamodel features to capture the meaning behind user's information needs and further to obtain a formal SPARQL query representation by means of model-to-model (M2M) transformation between the two.

3.1 Information Extraction

Information Extraction (IE) module aims to structure natural language document text at the level of ontological semantics, i.e. by analyzing entity mentions and their domain-specific relations we populate a predefined ontology schema with instance data. Formal ontological representation of document content allows taking advantage of implicit knowledge that can be inferred by employing OWL reasoning capabilities.

IE task behind our framework is powered by a pipeline of NLP components for Lithuanian language:

- Lexical analyzer performs stop word removal and standard text tokenization by breaking input text into words, sentences and paragraphs.
- Morphological analyzer assigns part-of-speech (POS) tags to each of the word along with lemma, grammatical number and most importantly grammatical case.
- Named Entity recognizer (NER) is based on gazetteer lookups. It detects mentions of entities that belong to three major type categories: organizations, locations and persons.
- Semantic annotator analyzes domain-specific relations between entities and produces ontology instance data in the form of RDF triples.

Each of the NLP components produces *stand-off* annotations in a custom data format which gets

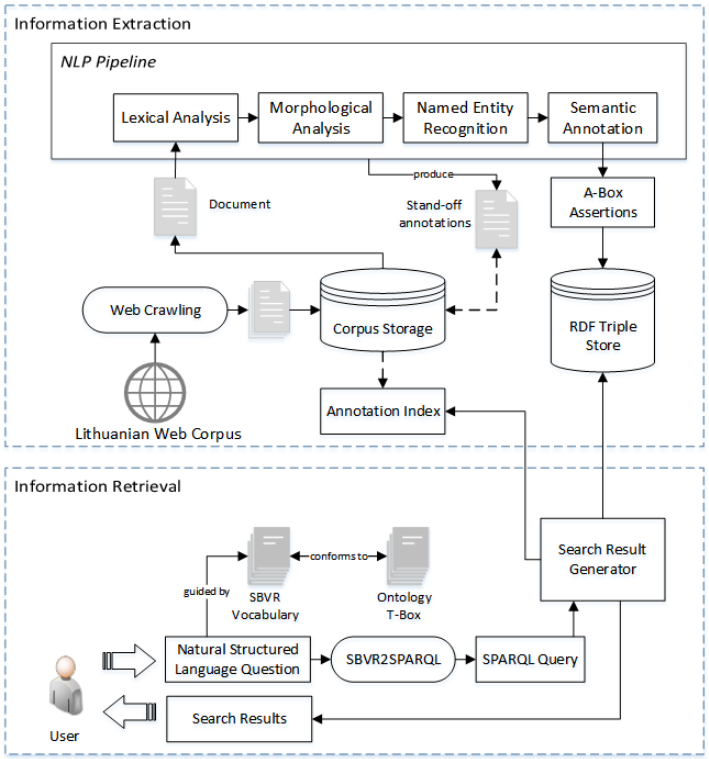

Figure 1: The architecture of the framework.

serialized using JSON. In such way, we keep the documents and their annotations decoupled.

Since the principle behind the three first NLP components in the pipeline is beyond the scope of this paper we will focus on the fundamental features of our semantic annotator.

Given an ontology schema, semantic annotator attempts to populate it by instantiating classes and their properties with entity and relation mentions found in the analysed text. It follows a rule-based approach that looks for specific lexico-semantic patterns, combining information from prior lexical, morphological and named entity annotations. Capturing all the domain-specific relations one could express in ontology is a non-trivial task, especially for highly inflected languages like Lithuanian. Moreover, the absence of production ready syntactic parsers makes the task even more challenging.

Our current ruleset targets extraction of political and economic event mentions in their various forms. We collected the most common reporting verbs (*sakyti* (say), *teigti* (state), *pranešti* (announce) etc.) from the news articles and derived multiple patterns for utterance extraction. Example rules are given below:

Rule I

```
  (c1)          (c2)     (c3)
{„SUBSTANCE“,-} <RVERB> <NE>
& type(NE) = Person =>
assert(c1:Substance, c2:Saying,
c3:Person, says<c3,c2>,
includes<c2,c1>)
```

Rule II

```
(c1)     (c2)                (c3)
<NE> <RVERB>{,} {kad|jog} {SUBSTANCE}
& type(NE) = Person =>
assert(c1:Person, c2:Saying,
c3:Substance, says<c1,c2>,
includes<c2,c3>)
```

Rule I is based on direct quotation extraction, while Rule II extracts indirect quotations by matching common conjunction *kad, jog* (that) patterns. In both cases we try to catch and instantiate the full triple: the agent, the reporting verb and the reported substance.

Some of the extraction rules are not as straightforward and require more attention to

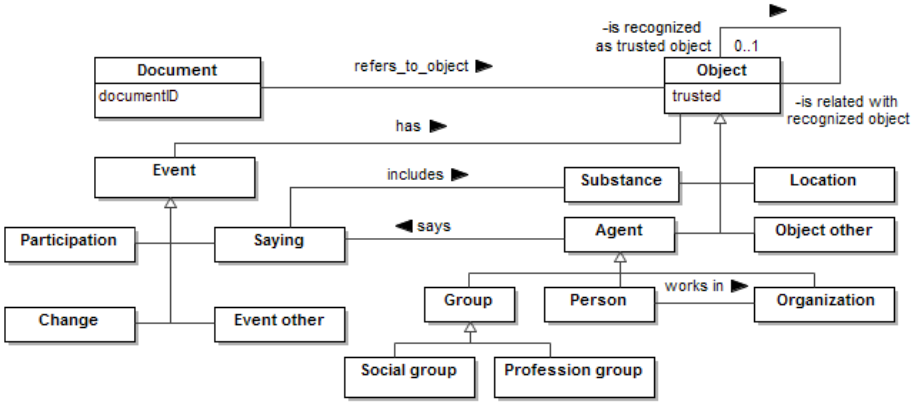

Figure 2: A fragment of domain-specific event ontology.

language specific features. An example of this is the detection of positions held by persons within organizations (here, PNOUN stands for the position noun like *prezidentas* (president), *ministras* (minister), *teisėjas* (judge) etc.):

Rule III

```
(c1)     (c2)    (c3)
<NE1> <PNOUN> <NE2>
& type(NE1) = Organization &
type(NE2) = Person
& caseMark(NE1) = genitive =>
assert(c1:Organization, c3:Person,
works in<c3,c1>)
```

By relying solely on lexical term sequence, we could easily end up with many incorrect extractions. In a sample sentence *Europos Parlamente prezidentė Dalia Grybauskaitė skaitė pranešimą* (President Dalia Grybauskaitė gave a speech at the European Parliament) the locative case of the word *Europ**os*** *Parlament**e*** determines its grammatical function - an adverbial modifier of place. Ignoring the case mark, Rule III would result in assertion `works_in<Dalia Grybauskaitė, European Parliament>` which is not entirely true. Therefore, an additional check for the genitive case is made to avoid incorrect extractions caused by Lithuanian declension.

Among the three rules presented above, our ruleset includes over 20 patterns for detecting changes of prices, taxes and other abstract objects of interest. Also, we always instantiate named entity mentions, whether they participate in some event or not.

The final assertions are produced according to the ontology schema that we created for capturing the event-specific knowledge commonly found in Lithuanian news articles. Currently, it consists of over 100 classes and nearly 70 relations. A tiny fragment of the ontology relative to the running examples throughout the paper is presented in Figure 2. The link between the document and the recognized objects within the content is established by an object property `<:refers_to_object>`. The `Object` class is the top class of all domain entities that we try to detect through the IE process. Thus, the enrichment of ontology with new domain entities is only a matter of sub-classing `Object`.

As every mention of a named entity within the text results in new instance (and URI) creation, we face ambiguity issues. The same entities tend to be referred to under different lexical aliases (*Dalia Grybauskaitė*, *D. Grybauskaitė*, *Grybauskaitė* etc.) throughout the news articles. Lithuanian declension causes even more suffix alternations (*Daliai Grybauskaitei*, *Dalią Grybauskaitę*, *Dalios Grybauskaitės* etc.) in such way having a negative impact on recall with queries including proper names (see Section 3.2). This is approached by employing several heuristics to disambiguate all the different entity mentions to a single entity we call *trusted*:

- First, we find equal entities by a common lemma and abbreviation matches.
- Secondly, we determine the main alias behind the *trusted* entity by inflecting its nominative case, and then create a new instance *T*.
- Lastly, we link all the corresponding entities to the *trusted* instance *T* by an object property `<:recognized_as_trusted_object>`.

We iterate the above process for each distinct entity type (organizations, locations, persons) recognizable by NER. In addition, the ontology is pre-populated with a set of well-known *trusted* entities along with their main aliases, which makes

the disambiguation process more precise.

As mentioned in Section 3, in addition to *A-Box* ontology assertions, semantic annotator produces stand-off *semantic annotations*, i.e. extracted document text fragments get linked to their corresponding ontology entity URIs by token indices. This information is later used in IR phase.

3.2 Information Retrieval

Our semantically enhanced Information Retrieval (IR) model builds upon the framework for querying OWL2 ontologies using structured natural language, as presented in (Sukys et al., 2012). The operation of this framework depends on SBVR, OWL2 and SPARQL specifications, each of which comes with a formal metamodel, thus making model-to-model transformations possible. SBVR business vocabulary is used to formulate, serialize and transform user's information needs to a SPARQL query which, eventually, retrieves appropriate answers from the ontology. An advantage of such model-driven IR approach is the ability to capture and map domain-specific business restrictions to formal query conditions (triple patterns) in a straightforward way, once the M2M transformation rules are present.

To ensure that the resulting triple patterns of SPARQL query correspond to ontology classes and properties, it is important to keep correspondence between SBVR vocabulary entries (general concepts, verb concepts) and OWL2 ontology entities. The most reliable way to do this is to obtain ontology schema automatically using model transformations from specifications of SBVR business vocabulary and business rules as described in (Karpovič et al., 2014) or vice versa (Bernotaityte et al., 2013).

Having business vocabulary and corresponding ontology schema in place, questions can be written using structured natural language, which helps to express the intents more precisely and avoid ambiguities that are common in natural language interpretation. Question formulation using structured natural language under strict grammar rules imposes the need to guide the end user throughout the process. Therefore, we employ EBNF (Extended Backus–Naur Form) to constraint user input to a somewhat relaxed form of possible SBVR formulations present in the vocabulary. As the question is written with the help of contextual suggestions, it gets parsed using EBNF rules and a syntax tree is created. The latter, which contains recognized statements of the question, is further used to generate SBVR XMI (*XML Metadata Interchange*) model. This model holds the captured meaning of a question that is constructed using a closed projection with restricting logical formulations and projection variables, expressing general concepts that should appear in the answer. SBVR XMI model is further transformed into SPARQL XMI model using ATL model-to-model transformation language. The principles behind transformation process and specific transformation rules are described in more details in (Sukys et al., 2012). At the final step, SPARQL XMI model is translated to textual representation using a model-to-text generator.

An illustrative example of transforming question "*Kokie asmenys dirba organizacijose?"* (What persons work in organizations?) to a SPARQL query is given below. For the sake of simplicity, we provide only a small fragment of SBVR vocabulary (Lithuanian and English equivalents), necessary for such transformation:

```
asmuo
organizacija
asmuo dirba organizacijoje
```

```
person
organization
person works in organization
```

The fragment consists of three vocabulary entries: general concepts `asmuo` (`person`), `organizacija` (`organization`) and a verb concept `asmuo dirba organizacijoje` (`person works in organization`) denoting the domain specific relation between the prior defined concepts. The declension of Lithuanian nouns can be clearly seen from the above example, i.e. the word representing general concept `organizacija` changes its suffix in the verb concept `asmuo dirba organizacijoje` since the verb *dirba* governs the locative case. In the English example, the grammatical form of a general concept `organization` remains the same since its role is determined by the use of the preposition *in*. We manage such language inflection by referring to the same concepts in different SBVR formulations by their main grammatical form – lemma.

Several heuristics are employed to make the structured question as natural sounding as possible. For example, in certain cases we allow omitting the subject part of the question, which is later derived by performing grammatical case-based matching in the vocabulary entries. As a result, the question in our running example can be expressed in a more user-

friendly form "*Kas dirba organizacijose*?" (Who work in organizations?). Similarly, we manage singular and plural noun forms as well.

The textual representation of a transformed SPARQL query from the question in our running example (English equivalent) is presented below:

```
SELECT ?person_i ?organization_i
WHERE {
  ?person_i ?person_in_organization
  ?organization_i .
  ?person_in_organization :label
  "person works in organization".
  ?person_i rdf:type ?person_cl.
  ?person_cl :label "person".
  ?organization_i rdf:type
  ?organization_cl.
  ?organization_cl :label
  "organization".
}
```

After the initial transformation, SELECT clause projects a set of variables V that bind (given the RDF graph data matches) to answer-bearing ontology entity URIs. The basic graph pattern (BGP) consists of multiple triple patterns that reflect the identification of conforming vocabulary and ontology concepts. In particular, we determine the type of each of the projected variables $v \in V$ in two steps:

- A triple pattern *T1* is created that binds a representative literal value of SBVR concept to a non-projected variable *n* (`<?person_cl :label "person">`).
- A subsequent triple pattern *T2* is created with *n* in an object position denoting the type of the projected variable *v* (`<?person_i rdf:type ?person_cl>`).

In a similar way we identify necessary vocabulary-conforming ontology properties.

Queries with proper names involved, e.g. *Kas dirba Europos Parlamente*? (Who works in the European Parliament?), are transformed by additionally employing simple heuristics to retrieve disambiguated instances (see Section 3.1). In particular, we generate a set of triple patterns that use the `<:recognized_as_trusted_object>` predicate to bind to all the *non-trusted* instances, thus giving higher recall.

Finally, the query is augmented with triple patterns that require for each of the projected variables to be bound to a single document instance (variable *d*), i.e. for each $v \in V$ we create triple patterns `<d :refers_to_object v>`. At the last step, we project an additional variable *k* in the SELECT clause that denotes the internal document identifier later on used for snippet generation. Note that $k \notin V$. An ORDER BY clause could be added to sort the results according to document publication date however, the ordering cost proved to be too high on a larger dataset.

At this stage, we have fully-constructed a formal SPARQL query that returns entity URI bindings, essentially performing *data* retrieval. With the original research aim in mind to attempt meaning-based *information* retrieval, our proposed framework includes a component for result snippet generation. The logic behind it is based on the following algorithm:

- For each of the initial SPARQL projection variables $v \in V$ extract their URI bindings $v \rightarrow u$;
- For each u retrieve its beginning b and ending e token indices from the semantic and lexical annotations produced in IE phase;
- Calculate $min(b)$ and $max(e)$ values to determine the range of a text passage;
- If $min(b)$ and $max(e)$ fit within boundaries of a single sentence, extend the range of a text passage to a full sentence;
- Else If $min(b)$ and $max(e)$ overlap to the neighboring sentences, extend the range of a text passage to the boundaries of neighboring sentences;
- Extract the text passage as a final snippet.
- Repeat for every tuple in the binding set.

Given that the lexical and semantic annotations produced in IE phase are correct, the above algorithm results in a snippet containing both the answer-bearing entities, and the original context they were extracted from.

4 EVALUATION

An early evaluation of our approach was performed by conducting a case study over a crawled corpus of Lithuanian news texts. We gathered over 90.000 domain specific documents from more than 30 news portals. After initial pre-processing steps the documents were annotated producing around 44 million explicit and 49 million implicit RDF triples under OWL-Horst materialization settings in the triple store. A prototype for the search interface was deployed to ease the evaluation of the practical applicability of our approach (see Figure 3).

In order to evaluate the search results in a quantitative manner, we selected 4 different queries for accuracy calculations: 2 abstract ones and 2 with

Figure 3: A prototype of semantic search interface for the case study. Sample results are shown for a question *Kas dirba Europos Parlamente?* (Who works in the European Parliament?).

proper names involved (see Table 1). We then judged the quality of the results on two main criteria: whether the text snippet returned gives a correct answer to the original question and if the answer-related entities are correctly highlighted within the text passage.

As a single article could possibly contain multiple distinct answers to the same query, we chose to calculate precision values snippet wise, so the total number of analysed articles differs per query and ranges from 12 to 65. Queries Q1, Q2 and Q3, Q4 were assessed by manually evaluating 61 and 71 snippets respectively. These numbers proved to be enough to observe a general trend in error sources.

Getting correct recall values is not a straightforward task in our current setting since a full set of correct answers to each of the queries is not known in advance. Therefore, we calculated recall only on a working subset of articles, i.e. those that had their snippets evaluated as mentioned above. In particular, we analysed the content of those articles to collect the number of missed annotations and assertions required to stand as an additional answer (snippet) with respect to the original query.

Table 2 shows the primary results of text snippet evaluation. Here, A_F column stands for the amount of snippets analysed, A_{FC} – snippets with correct answer, A_{NF} – not found snippets. While most of the queries achieve very high precision rates, Q2 stands out with a bit lower results. We noticed that a common pitfall here is the extraction of indirect quotations, where pure lexico-semantic patterns can't differentiate between the reporting agent and other agents contextually related to the reported substance. Relatively low recall values indicate that our current domain-specific event extraction ruleset is capable of capturing only the most common event expressions.

Table 1: Query set used for evaluation.

#	Query
Q1	*Ką kalbėjo agentai?* (What did the agents say?)
Q2	*Ką kalbėjo Vladimiras Putinas?* (What did Vladimir Putin say?)
Q3	*Kas dirba organizacijose?* (Who work in organizations?)
Q4	*Kas dirba Europos Parlamente?* (Who works in the European Parliament?)

Table 2: Text snippet accuracy results.

#	A_F	A_{FC}	A_{NF}	Recall	Precision
Q1	61	57	64	0.456	0.934
Q2	61	54	50	0.486	0.885
Q3	71	69	59	0.493	0.971
Q4	71	71	16	0.816	1.000

In addition, we evaluated accuracy of entity highlighting within the correctly returned snippets (see Table 3). Each of the queries from our query set is expected to return an entity tuple, either *agent-substance* (Q1, Q2) or *person-organization* (Q3, Q4), hence we split the results by agent/person and substance/organization columns. A_{FC} column lists the total number of snippets analysed, A_{AP} – correctly highlighted agent/person entities, A_{SO} – correctly highlighted substance/organization entities. As the results in Table 3 show, Q2, Q3 and Q4 reach

near-perfect precision values in both A_{AP} and A_{SO}. This is because these queries mostly return entities that get instantiated by strictly following NE tags. In contrast, Q1 fails significantly on A_{AP}. The *Agent* entity behind Q1 is more general according to our ontology schema (see Figure 2), therefore not only NE instances are included within the results. In particular, *Agent* subclasses like *Group* (and other specific types of agents) get instantiated by domain list-guided noun lookups during IE. Our efforts here fail short when the looked-up noun only governs the correct entity to be instantiated in a noun phrase and does not stand as an instance by itself. Thus, noun phrase mining should be improved by taking into account more Lithuanian morphological features.

Table 3: Entity highlighting accuracy results.

#	A_{FC}	A_{AP}	A_{SO}	P_{AP}	P_{SO}
Q1	57	36	53	0.632	0.930
Q2	54	54	51	1.000	0.944
Q3	69	69	69	1.000	1.000
Q4	71	71	71	1.000	1.000

The primary experimental evaluation of our approach led to certain observations:

- The precision of search results is mainly affected by the performance of NLP components behind IE task, since the IR phase operates in a Boolean manner, i.e. given a transformed formal SPARQL query the returned bindings hold all the conditions expressed by the set of triple patterns.
- Syntactic parser for Lithuanian is a crucial linguistic component currently missing from the NLP pipeline. Event extraction from complex sentence structures with a free word order is a non-trivial task and can hardly be carried out by solely relying on lexico-semantic patterns.
- Even when NLP-related errors occur at IE stage, our snippet generation approach enables to deliver a correct text passage with a decent level of accuracy.

The evaluation results can be summed up on a qualitative note. As shown in Figure 3, our strategy for snippet generation attempts to present the user with an answer in a single step, eliminating the need to perform an additional search for the answer-related text passage by opening the whole article. We see this as one of the main features and advantages of *semantic search* paradigm when compared to classical pure keyword-based approaches.

5 CONCLUSIONS AND FUTURE WORK

We presented, to the best of our knowledge, the first public attempt to semantic search over Lithuanian language texts. The aim of our research was to show that meaning-based information retrieval methods can be successfully applied even for resource-poor, highly inflected languages like Lithuanian. While state-of-the-art in Lithuanian natural language processing is far away from well computerized languages such as English, information extraction at the level of ontological semantics can still be approached with significant results. However, language specific linguistic features should be stressed.

The semantic search framework we proposed is a continuation of our previous research on semantics of SBVR, OWL2 and SPARQL. In particular, we study the conceptual conformity and discrepancies between their metamodels by means of model-to-model transformations. Although the practical application of our model-driven approach requires some customizations by preparing business vocabularies and ontologies, it can be ported to different domains. SBVR question transformation to SPARQL is not dependent from the structured language used. However, certain adoptions for using different languages, such as Structured English or Structured Lithuanian are needed due to grammatical peculiarities of languages.

The early case study demonstrated promising results and thus we seek to improve our efforts, especially towards content processing. By refining event extraction rules and patterns we hope to capture more domain-specific event mentions along with their event-specific characteristics. Moreover, we plan on evaluating the gathered RDF knowledge base in order to draw some statistical conclusions regarding the most common events and entities found in the news articles. With such information available, we will aim at performing fine-grained ontological typing of the most common entities against public knowledge bases, by employing named entity linking techniques. Finally, our prototype from the case study, as a part of a bigger project, is being launched for public availability with the hope to provide better news search capabilities to Lithuanian language users. See: http://www.semantika.lt/SemanticSearch/Search/Index

REFERENCES

Salton, G., Wong, A., Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM, 18(11), 613-620.*

Carpineto, C., Romano, G. (2012). A survey of automatic query expansion in information retrieval. *ACM Computing Surveys (CSUR),* 44(1), 1.

Stokoe, C., Oakes, M. P., Tait, J. (2003). Word sense disambiguation in information retrieval revisited. *In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 159-166). ACM.*

Mangold, C. (2007). A survey and classification of semantic search approaches. *International Journal of Metadata, Semantics and Ontologies*, 2(1), 23-34.

Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A., Welty, C. et al. (2010). Building Watson: An overview of the DeepQA project. *AI magazine*, 31(3), 59-79.

Šveikauskienė, D., Telksnys, L. (2014). Accuracy of the Parsing of Lithuanian Simple Sentences. Information Technology and Control, 43(4), 402-413.

Kiryakov, A., Popov, B., Terziev, I., Manov, D., Ognyanoff, D. (2004). Semantic annotation, indexing, and retrieval. Web Semantics: Science, Services and Agents on the World Wide Web, 2(1), 49-79.

Castells, P., Fernandez, M., and Vallet, D. (2007). An adaptation of the vector-space model for ontology-based information retrieval. Knowledge and Data Engineering, IEEE Transactions on, 19(2), 261-272.

Fernández, M., Cantador, I., López, V., Vallet, D., Castells, P., Motta, E. (2011). Semantically enhanced Information Retrieval: an ontology-based approach. Web Semantics: Science, Services and Agents on the World Wide Web, 9(4), 434-452.

Lopez, V., Uren, V., Sabou, M. R., Motta, E. (2009). Cross ontology query answering on the semantic web: an initial evaluation. *In Proceedings of the fifth international conference on Knowledge capture (pp. 17-24). ACM.*

Zinkevičius, V. (2000). Lemuoklis–morfologinei analizei. Darbai ir dienos, 24, 245-274.

Šveikauskienė, D. (2005). Formal description of the syntax of the Lithuanian language. Information Technologies and Control, 34(3).

Kapociute-Dzikiene, J., Nivre, J., Krupavicius, A. (2013). Lithuanian Dependency Parsing with Rich Morphological Features. *In Fourth Workshop on Statistical Parsing of Morphologically Rich Languages (p. 12).*

Krilavičius, T., Medelis, Ž., Kapočiūtė-Dzikienė, J., Žalandauskas, T. (2012). News Media Analysis Using Focused Crawl and Natural Language Processing: Case of Lithuanian News Websites. In Information and Software Technologies (pp. 48-61). *Springer Berlin Heidelberg.*

Amardeilh, F. (2008). Semantic annotation and ontology population. Semantic Web Engineering in the Knowledge Society, 424-p.

Navigli, R., Ponzetto, S. P. (2012). BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence, 193, 217-250.*

OMG, 2008. Semantics of Business Vocabulary and Business Rules (SBVR). Version 1.0. December, 2008, OMG Document Number: formal/2008-01-02.

Goedertier, S., Vanthienen, J. (2008). A Vocabulary and Execution Model for Declarative Service Orchestration. *Business Process Management Workshops, LNCS, Vol. 4928, 496–501.*

Bodenstaff, L., Ceravolo, P., Ernesto Damiani, R., Fugazza, C., Reed, K., Wombacher, A. (2008). Representing and Validating Digital Business Processes. Web Information Systems and Technologies, LNBIP, Vol. 8(1), 19–32.

Karpovič, J., Kriščiūnienė, G., Ablonskis, L., Nemuraitė, L. (2014). The Comprehensive Mapping of Semantics of Business Vocabulary and Business Rules (SBVR) to OWL 2 Ontologies. Information Technology and Control, 43(3), 289-302.

Sukys, A., Nemuraite, L., Paradauskas, B., Sinkevicius, E. (2012). Transformation framework for SBVR based semantic queries in business information systems. In BUSTECH 2012, *The Second International Conference on Business Intelligence and Technology (pp. 19-24).*

Sukys, A., Nemuraite, L., Paradauskas, B. (2012). Representing and transforming SBVR question patterns into SPARQL. *In Information and Software Technologies (pp. 436-451).*

Bernotaityte, G., Nemuraite, L., Butkiene, R., Paradauskas, B. (2013). Developing SBVR vocabularies and business rules from OWL2 ontologies. *In Information and Software Technologies (pp. 134-145).*

Shekarpour, S., Marx, E., Ngomo, A. C. N., & Auer, S. (2015). Sina: Semantic interpretation of user queries for question answering on interlinked data. Web Semantics: Science, Services and Agents on the World Wide Web, 30, 39-51.

Yao, X., Van Durme, B. (2014). Information extraction over structured data: Question answering with freebase. *In Proceedings of ACL.*