Text Analysis with Ontology Reasoning

Anna Rozeva
Department of Computer Systems and Informatics, University of Forestry, Kliment.Ohridski blv.10, Sofia, Bulgaria
arozeva@hotmail.com

Keywords: Text Analysis, Text Mining, Domain Ontology, Context Model, Ontology Reasoning, Knowledge Management.

Abstract: The analysis of unstructured text when performed by text mining machine learning algorithm results in mining model holding rules for relationships and dependencies among terms extracted by text pre-processing techniques. The obtained mining model represents knowledge derived from the analyzed text which is hard to interpret as it lacks context. Enhancement of its semantic value can be obtained by implementing logic based approach providing formally defined meaning and interpretation mechanism. The generally accepted form for representation of knowledge for existing domain is by domain-specific ontology. The aim of the paper consists in designing a framework for performing ontology instantiation and population with the structures of a complex mining model involving classification and association rules. Procedures have been designed for annotating them with domain concepts and semantic types. The framework provides for turning the mining model into a context model. Ontology reasoning is implemented to validate the input mining model by rule semantic disambiguation and dependency conceptualization. The framework implementation provides for outputting validated domain-related knowledge base in explicit and machine-readable form as a resource that can be adapted for decision support.

1 INTRODUCTION

Text analysis has become a topic attracting increasing research interest and efforts in the recent decade. The reason consists in the enormous and continuously growing amount of published information on Internet. Searching for specific subject area related documents more efficiently as compared to the traditional keyword-based techniques has emerged as one of the most challenging and value-adding problems concerning it. Classification of text documents has turned out to be the most important task in text analysis so far as the retrieval of information from the web is concerned. The text analysis performed by implementing classical syntactical and statistical text mining techniques results in mining models that generally lack sense and ignore the semantics of the input documents.

Most recent researches present techniques and approaches for ensuring the lacking semantic integration in a text mining model and improving machine-learning based text classification (Albitar, Fournier and Espinasse, 2012; Garla and Brandt, 2012). They implement semantic resources like thesauri and ontologies for text "conceptualization" (Albitar et al., 2012) or taxonomical structures for improving feature ranking and semantic similarity measures for projecting text into a feature space (Garla et al., 2012) thus enriching the semantics of the classification output.

Machine accessible semantics of text mining models is achieved by means of ontologies (Horrocks, 2008). They represent hierarchically structured conceptual schemas that give formally defined meanings to concepts referring to a specified domain. As a model of a piece of the world it provides vocabulary describing the aspects of the modelled domain as well as an explicitly stated meaning of its content. The meaning generally represents classification information.

Ontology-based methods have been implemented for refining text analysis and information extraction process. Extraction of links between concepts for capturing domain semantics by using domain specific ontology has been considered in (Morneau and Mineau, 2008).

Ontologies provide semantic frameworks for the interpretation of analyzed information. This becomes possible by using logical formalisms for their representation. Description logic (DL) (Baader

Rozeva A.
Text Analysis with Ontology Reasoning.
DOI: 10.5220/0004774100640073
In *Proceedings of the Third International Symposium on Business Modeling and Software Design (BMSD 2013)*, pages 64-73
ISBN: 978-989-8565-56-3

et al., 2007) is a knowledge representation formalism distinguished by formal semantics and provision of inference services. It is the foundation of ontology languages like OWL DL (Motik et al., 2012). Ontology based on description logic provides reasoning tools and services for designing and maintaining qualitative ontologies, answering queries over its classes and instances, integrating and mapping ontologies. Therefore ontology reasoning facilitates the design and development of ontologies as well as their deployment in applications.

The intensively researched problem areas mentioned so far provide the motivation for the current work. The aim is to design a framework for implementation of domain ontologies and reasoning services in text document analysis enabling the design of qualitative and validated domain related knowledge. The rest of the paper is organized as folows: Section 2 examines approaches, platforms, mechanisms and applications of ontologically based text analysis; Section 3 presents the conceptual structure of our framework for ontological text analysis; Sections 4 and 5 highlight framework implementation issues and Section 6 concludes with discussion and directions for future work.

2 REVIEW ON ONTOLOGIES IN TEXT ANALYSIS

Text analysis is used in the sense of automatic processing of huge volumes of textual information in order to facilitate its retrieval, management and structuring for research purposes. On the other hand this notion concerns also the extraction of context and meaning from the processed text corpus. Each of these aspects is performed by the implementation of specific information technology.

The technology which converts unstructured or semi-structured natural language text in a form that is suitable for machine processing which results in models of extracted facts, discovered implicit links and generated hypotheses, is referred to as text mining. The models obtained by text mining represent knowledge which needs to be properly interpreted, shared and integrated. This can be accomplished by combining them with corresponding conceptual models which reflect the structure of the subject area the knowledge is referring to and defines the interpretations of its terms, i.e. ontologies. They specify concept meaning and their relations to other concepts in machine-readable form. Ontologies therefore provide for the automatic semantic interpretation of textual information and enhance the benefits of text mining. Further on text analysis will be distinguished as text mining analysis and ontological analysis.

Figure 1 shows a framework for text mining analysis performed on unstructured text corpus that is presented in (Rozeva, 2011a).

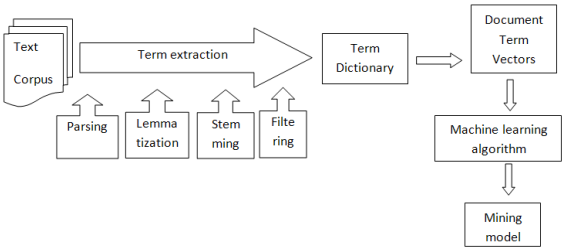

Figure 1: Text mining analysis framework.

Within the above shown framework text is lexically and syntactically processed for the extraction of descriptive terms. They are determined on the basis of evaluating frequency of appearance in a document. Different methods for frequency calculation can be implemented for enhancing the relevance of the extracted terms to the document content. The extracted descriptive terms provide for presenting documents by term vectors. Each document is represented by a vector, which contains the extracted terms and their frequency of occurrence in it. Obviously the document vector matrix is sparse. The matrix is processed with machine learning algorithms and in result mining model is obtained. It stores patterns in the form of classifications, groups and associations mined from the input documents.

Approach for implementation of text mining model within a specified platform for predicting unknown features of new documents is presented in (Rozeva, 2011b).

2.1 Ontological Text Analysis

Basic aspects and practices of using ontologies with text mining are reviewed in (Spasic et al., 2005). So far as ontology provides the terminology of a domain, terms are set to express specialized domain concepts. The mapping of term to concept is considered as the basis for the semantic interpretation of text mining models. Different layers of text annotation by means of ontologies are examined, i.e. lexical, syntactic and semantic.

A solution for the term ambiguity problem as a general text mining challenge is reported in (Bratus et al., 2009). It concerns cases when a term refers to

multiple concepts which are normal for natural language texts. The textual case-based reasoning system developed performs taxonomic indexing of text archives which provides for ontology-guided search. It requires text classification and disambiguation. The implemented classification method locates key phrases in lexical taxonomies. Disambiguation uses context to determine the taxonomy regions, which are likely to be most relevant. Algorithm has been designed for ontology-guided text disambiguation.

Use of ontologies during the text analysis process and as an export format for the results obtained is shown in (Witte et al., 2007). Software document ontology has been designed for being automatically instantiated by text mining module. The concepts to be included therein are the ones related to: document structure or lexica; lexical normalization rules; relations between classes and software specific entities. The text mining system for the instantiation of pre-modelled ontology (ontology population) has been designed within the GATE framework (Cunningham et al., 2011) for text engineering, language processing and text analysis. Entities resulting from natural language processing (NLP) populate the ontology which is further on processed for named entity recognition. Entities are matched against a list of terms and in case of match an annotation is added. An ontology-aware component incorporating mappings between term lists and ontology classes is used to assign the proper class in case of term match. Specially designed grammar rules are used to detect and annotate complex named entities. They refer to the annotations created and evaluate the ontology directly. They detect semantic units by combining ontology-based look-up information and extracted noun phrases from documents. By implementation of fuzzy-set based co-reference resolution system detected entities are grouped into chains. Normalization with a set of lexical rules provides entities with canonical names. Rules are stored in their corresponding classes in the domain ontology which allows the inheritance of rules through subsumption. Relations between entities are detected by grammar rules and syntactic analysis. The obtained relations are filtered through the ontology for discarding the semantically invalid ones.

Integration aspects of formal ontologies in OWL-DL format with text mining systems for supporting NLP are discussed in (Witte et al., 2007). Ontology is implemented both as a container for mining results and as a language for document processing. Requirements for the necessary information to be included in ontology for supporting text mining tasks are defined. Issues concerning interfacing ontologies to text mining systems have been considered as well.

Ontology population methods have been considered in (Wang et al., 2005) and (Damljanovic et al., 2009). Learning and populating ontology from linguistic resources is presented in (Wang, et al., 2005). Special attention is turned to the extraction of related concepts and the method applied prevents them to be far apart in the concept hierarchy. Creating adapted workflows for semi-automatically ontology population with text from diverse semantic repositories is reported in (Damljanovic et al., 2009).

Framework for ontology-based text categorization which performs ontology learning by mining input documents and uses the ontology further on for refining the categorization with both supervised and unsupervised machine learning approaches is presented in (Bloehdorn et al., 2005). The framework provides ontology learning algorithms such as for constructing taxonomies with conceptual clustering algorithm; for classifying instances into the ontology by using lexico-syntactic patterns; for extracting labeled relations and specifying their domain and range and for semantic enrichment by finding the appropriate concept from a given ontology, etc. The lexica and concept hierarchies of ontologies are involved in performing text clustering and classification by enhancing the common term representation of documents with concepts extracted from the used ontologies. Ontologies enable specific concepts found in text to be replaced by more general concept representations, i.e. the corresponding superconcept along the path to the root of the concept hierarchy. More recent review on attempts to combining machine learning techniques and ontologies by investigating mining of semantic web data sources with inductive learning techniques is presented in (Bloehdorn and Hotho, 2009).

2.2 Ontology Querying and Reasoning

Basic advantage of the implementation of formal ontologies in text analysis is that it enables the definition of semantic queries on concept instances and automated reasoning based on DL. Reasoning (Horrocks, 2008) has to check that ontology knowledge is meaningful, correct, minimally redundant, richly axiomatised, that queries can be answered over its classes and instances, that individuals matching a query can be retrieved and that the knowledge base is consistent.

Many inference tasks are reduced to subsumption reasoning and to satisfiability.

Efficiency of query answering reasoning task is discussed in (Pothipruk and Governatori, 2005). DL-based reasoning performs inference on a knowledge base consisting of terminological axioms (Tbox) and assertion axioms (Abox). Tboxes refer to the schema of concepts and Aboxes to the names of individuals. Reasoning on the terminological hierarchy, specified by DL first order formulas, checks the fulfillment of the following logical requirements: concept satisfiability; class subsumption; class consistency; instance checking.

A subset of reasoning rules (Wang et al., 2004) is shown in Table 1.

Table 1: Sample OWL ontology reasoning rules.

Transitive_ Property	(?P rdf:Type owl: TransitiveProperty)∧ (?A ?P ?B)∧ (?B ?P ?C)⇒ (?A ?P ?C)
subClassOf	(?a rdfs:subClassOf ?b) ∧ (?b rdfs:subClassOf ?c) ⇒ (?a rdfs:subClassOf ?c)
subPropertyOf	(?a rdfs:subPropertyOf ?b) ∧ (?b rdfs:subPropertyOf ?c) ⇒ (?a rdfs:subPropertyOf ?c)
disjointWith	(?C owl:disjointWith ?D) ∧ (?X rdf: type ?C) ∧ (?Y rdf: type ?D)⇒ (?X owl:differentFrom ?Y)
inverseOf	(?P owl:inverseOf ?Q) ∧ (?X ?P ?Y) ⇒ (?Y ?Q ?X)

The proposed optimization approach concerns Abox reasoning, as the basis for query answering. Two types of queries allowed by DL are considered. The Boolean query represents instance checking and the non-Boolean consists in retrieving Abox content. The approach addresses the problem of efficient answering a query in a DL-based semantic web system implementing single ontology and multiple data sources.

Architecture for ontology reasoning is proposed in (Pan, 2007). It supports ontology languages providing for the definition of customized data types and customized data type predicates. It allows new data type reasoners to be added into the architecture without affecting the basic concept reasoner.

Ontology for modeling context and supporting context reasoning has been designed in (Wang et al., 2004). Its upper level captures general concepts about the basic context. It allows hierarchical adding of domain-specific ontologies. Logic reasoning on this ontology consists in checking the consistency of context information and deriving high level implicit context from low level explicit context.

Ontologies provide the vocabulary that enables the semantic markup of web resources. Thus they express the terminological part of knowledge structured in taxonomy of concepts and properties. DL based language as OWL enables reasoning for ontology checking, classification and recognition of class instances. Rules on the other hand describe logical dependencies between the ontology elements and as such represent the deductive type of knowledge. Rule types are: standard – for chaining ontologies' properties; bridging – for reasoning across several domains; mapping – ontologies in data integration; querying – for expressing complex queries in ontology vocabulary and meta – for facilitating ontology engineering (acquisition, validation and maintenance).

As considered in (Golbreich, 2004) completeness of inferences can be obtained on the basis of the whole domain knowledge available both in ontologies and rule bases. Approach for reasoning by combining them is designed based on Semantic web standards language OWL and SWRL (Horrocks et al., 2004). The platform for combining DL and rule reasoning is shown in Figure 2.

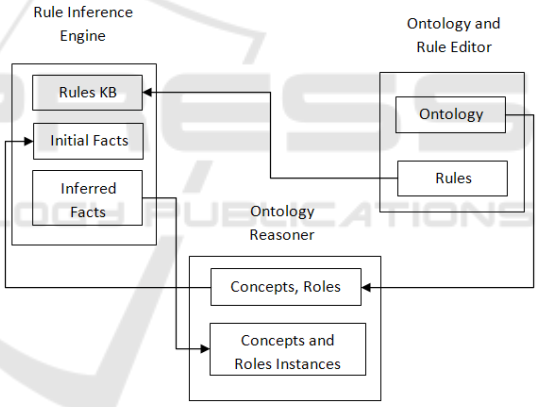

Figure 2: Combining ontology and rule reasoning.

Modeling approach for developing rule-based application for the web is presented in (Canadas, Palma and Tunes, 2009). It implements ontologies for describing the concepts and their relationships in the application domain and rules for formalizing the inference logic thus providing for increasing the amount of knowledge represented in ontologies.

3 CONTEXT-BASED TEXT ANALYSIS FRAMEWORK

The review of related work on approaches, methods, technologies and tools presented in the previous

section has motivated the design of a framework for analysing web text documents resulting in a knowledge base that can be queried with automatic semantic reasoners for inferring implicit facts. In our previous work (Rozeva, 2012) we've presented an approach for implementing pre-defined ontology at:

- Text pre-processing step for filtering terms that don't map to ontology individuals;
- Post-processing the rules learnt by implementing mining algorithm.

The approach implemented in the current work uses the rules model obtained by processing the text with machine learning algorithm for instantiation of domain ontology. This provides for enhancing the efficiency of the representation of the analysed domain by turning the mined model into a context model. The context model being ontologically based ensures logically validated classification and logic reasoning. The framework is shown in Figure 3.

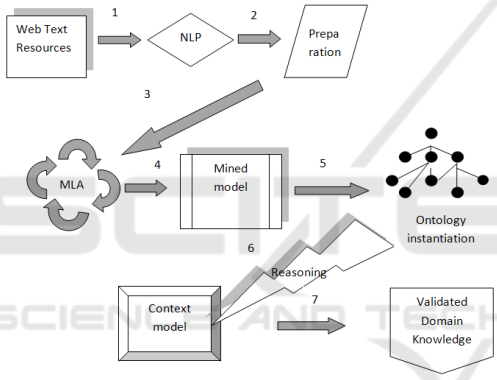

Figure 3: Ontological text analysis framework.

The modules in the framework are:

- NLP – natural language text processing resulting in term extraction;
- Preparation of document vector matrix for algorithmic processing;
- MLA - machine learning algorithm producing mined model;
- Ontology instantiation with the mined model producing semantically labelled context model;
- Ontology reasoning for inferring new facts resulting in validated domain related knowledge.

MLAs that are usually applied for text analysis purposes are: discovering groups of text documents by clustering; classification (grouping according to pre-defined categories) and associations.

The core of the proposed framework is the instantiation of ontology from the rules of mined model obtained by processing the text with machine learning algorithm. This process refers to defining classes, arrangement of classes in a taxonomic hierarchy, defining properties of classes and allowed values for them, filling in the property values for the individuals of the classes. The framework implements classification and association rules machine learning algorithms. The classification algorithm provides for building the ontology class hierarchy. The association rules algorithm enables the definition of non-taxonomic relations between classes. The method for ontology building is inspired by the method presented in (Elsayed et al., 2007). Their approach uses the rules obtained by mining structured data for mapping tree nodes to OWL classes, tree branches to OWL classes and leaf nodes to individuals. The ontology building algorithm implements functions for: getting the branches of a node and the branch of a leaf node; getting the class that represents a branch and for creating individual for a leaf node. This algorithm is adapted in the proposed framework for taking input from rules mined from text and is enhanced with the definition of additional objectType property of the classes. The method for enriching the ontology with non-taxonomic relations is based on similar research held on lexico-syntactic patterns from domain specific dictionary for extracting relations between concepts shown in (Maedche and Staab, 2000) as well as on an approach for mining dictionary databases for ontology generation purposes presented in (Deliyska et al., 2012).

3.1 Building Ontology Taxonomy

The proposed algorithm for ontology instantiation comprises two modules. The first one creates ontology from a decision tree mining model. The second one performs ontology enrichment by creating relations between ontology classes from the results of an association rules model. The general structure of decision tree model shown in Figure 4 provides the input for the ontology building module, i.e.:

- Predictable attribute name;
- Node ID and node name;
- Node type – root, interior or distribution;
- Node children cardinality – 2 or 0;
- Node parent;
- Node description – inputAttributeValue_ missing or inputAttributeValue_existing;
- Node rule, containing attribute and predicate values;
- Node distribution – probability of predictable

attribute values.

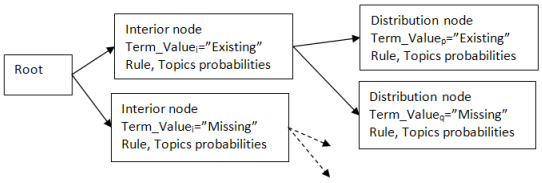

Figure 4: Decision tree text analysis model.

The predictable attribute in a text classification task is the document category (Topic). The tree is split by the presence or absence of an input attribute value. The input attribute in the text classification is Term with values extracted in the NLP stage. Cardinality for interior nodes is two, denoting existing or missing term value. Each node has probability distributions of predictable attribute values attached. The algorithm steps for mapping the decision tree to ontology components are:

Step 1: Define independent class for the distribution with classification topics and probability;

Method:

```
Class Distribution = new(OWL:class)
Distribution.Id= Distribution.name
DatatypeProperty DistributionDP=new
(owl:DatatypeProperty)
Class Topic = new(OWL:class
subClassof Distribution)
Topic.Id= Topic.name
DatatypeProperty TopicDP=new
(owl:DatatypeProperty)
Class TopicProbability =
new(OWL:class subClassof
Distribution disjointWith Topic)
TopicProbability.Id=
TopicProbability.name
DatatypeProperty
TopicProbabilityDP=new
(owl:DatatypeProperty)
```

Step 2: Define independent class for the nodes' rules;

Method:

```
Class NodeRule = new(OWL:class)
NodeRule.Id= NodeRule.name
DatatypeProperty NodeRuleDP=new
(owl:DatatypeProperty)
Class Predicate = new(OWL:class
subClassof NodeRule)
Predicate.Id= Topic.name
DatatypeProperty NodeRuleDP=new
(owl:DatatypeProperty)
Class TermValue = new(OWL:class
subClassof NodeRule disjointWith
Predicate)
TermValue.Id= TermValue.name
DatatypeProperty TermValue DP=new
(owl:DatatypeProperty)
```

Step 3: Define independent class for the nodes' probability;

Method:

```
Class Probability = new(OWL:class)
Probability.Id= Probability.name
DatatypeProperty ProbabilityDP=new
(owl:DatatypeProperty)
```

Step 4: Define the root class for the taxonomy of the interior and terminal nodes;

Method:

```
Class All = new(OWL:class
disjointWith Distribution
disjointWith NodeRule disjointWith
Probability)
All.Id= All.name
DatatypeProperty All=new
(owl:DatatypeProperty)
```

Step 5: Define the class taxonomy from the interior and terminal nodes;

Method:

```
BEGIN
 For each node DN
 Class C=new (owl:Class)
 C.Id= DN.name
 DatatypeProperty DP=new
(owl:DatatypeProperty)
Dp.Id= DN.name+"_Value"
Dp.AddDomain(C)
     For each ChildNode CN of Get-
Children(DN)
 Dp. AddDomain (CN.Get-Class ())
     endfor
 endfor
End
```

Step 6: Define properties and make the relation of the class taxonomy to the independent classes;

Method:

```
ObjectProperty hasDistribution =
new(Owl:FunctionalObjectProperty)
ObjectProperty hasProbability =
new(Owl:FunctionalObjectProperty)
ObjectProperty hasRule =
new(Owl:FunctionalObjectProperty)
AddSubclassOf(All, hasDistribution)
AddSubclassOf(All, hasProbability)
AddSubclassOf(All, hasRule)
```

Step 7: Create individuals for ontology classes;

Method:

```
BEGIN
For each node DN
 Individual I = new
(owlclass:Individual)
     Foreach Topic T
Distribution.I+=
 T+TopicProbability
     Endfor
 NodeRule.I+= Predicate+TermValue
 ADDType(Distribution.I, I)
 ADDType(NodeRule.I, I)
Endfor
End
```

The ontology instantiated by this module involves the hierarchical relations between the defined classes. By including non-hierarchical relations as well further enrichment and refinement of the generated ontology will be obtained.

3.2 Ontology Enrichment

Ontology enrichment is achieved by the second module in the framework which introduces non-hierarchical relations between ontology classes. These relations are extracted by applying the association rules algorithm of Srikant and Agrawal (as cited in Maedche and Staab, 2000) on the processed text documents. Association rules are in the form X →Y with measures for confidence and support, where 'X' and 'Y' represent items in a transaction set.

When applied to hierarchically structured ontology classes the right side of the rule involves the ancestors of the particular item as well. The resulting rules contain 'Y' that isn't ancestor of 'X' and the rule X→Y isn't subsumed by one involving their ancestors. The association rules produced by the text mining algorithm are to be implemented as properties between ontology classes' individuals. The algorithm implements a method which assumes that individual corresponding to antecedent 'X' of the association rule exists in ontology. It involves the following method:

Method:

```
For each rule X→Y
If exist(owlclass.Individual="Y")
Begin
 ObjectProperty Prop_Name =
 new(Owl:ObjectProperty)
 AssertObjectProperty (X,
 Prop_Name, Y)
End
ElseIf
Begin
 Class Association = new(OWL:class
 disjointWith class.X)
 Association.Id= Association.name
 DatatypeProperty Association =new
 (owl:DatatypeProperty)
 Association:Individual="Y"
 AssertObjectProperty (X,
 Prop_Name, Y)
End
Endif
EndFor
```

The method checks the presence of an individual 'Y' in ontology and creates object property which relates it to the individual 'X'. If not present new class is added to ontology as disjoint with the class individual 'X' belongs to. Individual 'Y' is added to this class. Object property is created and is asserted to relate individuals 'X' and 'Y'.

4 FRAMEWORK IMPLEMENTATION

The text mining module of the framework has been implemented in Microsoft SQL Server 2008 (Microsoft SQL Server, 2013). The classification model obtained by mining the text with Microsoft Decision Tree algorithm is shown in Figure 5.

NodeName	00000000000
NodeCaption	(e-governance) = Missing
NodeType	3 (Interior)
ChildrenCardinality	2
ParentNode	000000000
NodeRule	predicate, attribute
MarginalRule	predicate, attribute
NodeProbability	0,755555555555556
MarginalProbability	0,871794871794872
NodeDistribution	Topic Distribution
NodeSupport	34

Figure 5: Decision tree mining model.

Topic's distribution created for each tree node is shown in Figure 6. Term extraction and term lookup transformations have been implemented for text preparation. The document corpus consists of conference papers on e-Governance.

ATTRIBUTE_NAME	ATTRIBUTE_VALUE	SUPPORT	PROBABILITY	VARIANCE	VALUETYPE
Topic	Missing	0	0	0	1 (Missing)
Topic	A	6	0,184210526315789	0	4 (Discrete)
Topic	B	5	0,157894736842105	0	4 (Discrete)
Topic	C	10	0,289473684210526	0	4 (Discrete)
Topic	D	13	0,368421052631579	0	4 (Discrete)

Figure 6: Classification topic distribution.

For instantiating the ontology the approach for mapping database schema to ontology presented in (Yankova et al., 2008) has been implemented. The general form of the script statement mapping table column to ontology classes is:

```
_columnToURI.putForward("Table.Field",
"Ontology:hasField)
```

The instantiated ontology with the decision tree model by implementing the algorithm in Protégé 4.2 (Protégé 4 User Documentation, 2013) is shown in Figure 7.

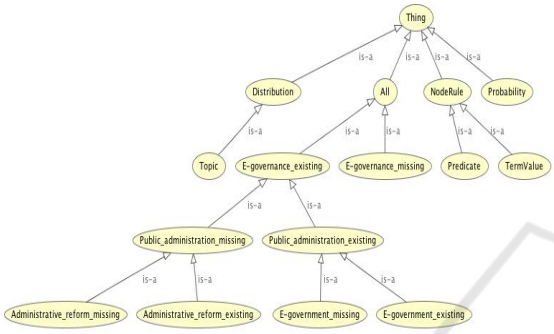

Figure 7: Ontology instantiated with decision tree model.

The association rules mining model obtained by processing the text corpus with the Microsoft Association Rules algorithm is shown in Figure 8.

Support	Size	Itemset
2	2	EU E-GOVERNANCE = Existing, e-governance center = Existing
2	2	public sphere = Existing, public service = Existing
2	2	public service = Existing, public administration = Existing
2	2	knowledge economy = Existing, public sector = Existing
2	2	civil servant = Existing, public service = Existing
2	2	public management = Existing, public administration = Existing
2	2	public management = Existing, public service = Existing
2	2	company management = Existing, quality management = Existing
1	2	administrative reform = Existing, e-government = Existing
1	3	administrative reform = Existing, public sphere = Existing, e-government = Existing
1	3	administrative reform = Existing, public service = Existing, e-government = Existing
1	3	administrative reform = Existing, public administration = Existing, e-government = Existing

Figure 8: Association rules mining model.

The ontology enriched with the mined rules is shown in Figure 9. The rule "administrative reform" = Existing → e-government = Existing has been inserted in ontology as shown with arrows by:

- Creating new class Reform;
- Creating individuals for the class, i.e. Administrative, Management and Budget;
- Creating object property "involves";
- Asserting the "involves" property to "e-government" individual of class "E-government_existing" with the individual "Administrative" from the Reform class.

The ontology instantiated by the proposed framework has been classified with Pellet reasoner (Pellet Reasoner Plug-in for Protégé 4, 2013) – Figure 10.

Figure 9: Enriched ontology with association relations.

Figure 10: Classified ontology.

5 REASONING EXAMPLES

A classified ontology can be searched by logical queries. Figure 11 presents sample DL query for retrieving class individuals with restriction on dataType property. Class queries get subclasses or descendents in the hierarchy or the superclass of a class.

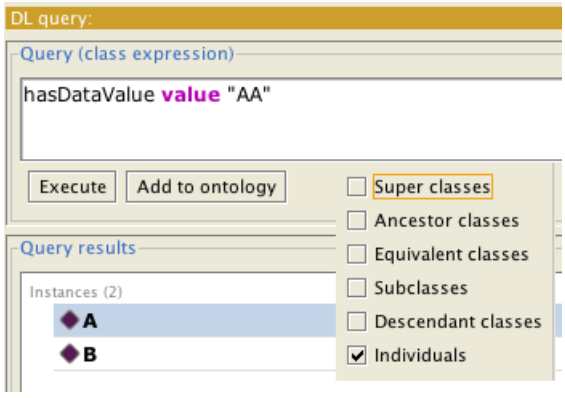

Figure 11: DL query on classified ontology.

Figure 12 presents sample OWL2 query in graph view. The TBoxes, ABoxes and RBoxes are used to provide concept relations, relations between individuals and concepts and rules to the query.

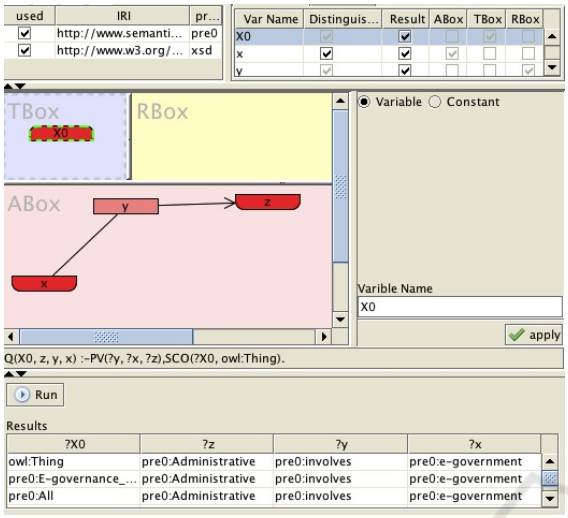

Figure 12: Owl2 Query on classified ontology.

6 CONCLUSIONS

A framework for text analysis involving the building and instantiation of ontology from analysis model obtained by text mining has been proposed. It has been designed with the aim of enhancing the semantic content of analysis model rules and turning them into a knowledge rule base which enables processing with logic reasoning. Most of the reviewed work on ontologies in text analysis considers natural language processing and validations with dictionaries and domain ontologies. Current approach addresses algorithmically processed text. It is considered that ontology obtained from mining model can be treated as more relative representation of text corpus overall content. By structuring a rule model into ontology the proposed framework ensures model's semantic enrichment. The contribution of the paper is the defined approach for representing a complex text mining model as concise domain ontology. The examined mining model is complex because it contains both classification and association rules. Thus ontology proves to be the means for integrating knowledge rules from different types of machine learning text processing. It also provides the processing tools for extracting the logical content and meaning of the integrated rules.

The framework couples a text mining and ontology instantiation modules. Methods are created for the automatic mapping of rules into ontology elements – classes, object and data properties and instances. The framework has been implemented for classification and associations analysis of selected text corpus. Classification as the most typical analysis task produces rules that enable the natural mapping to ontology class hierarchy. The framework implementation resulted in ontology that has been successfully classified by a reasoner. Examples for searching the classified ontology with description logic and OWL2 queries have been provided. It's claimed that the ontology mapping besides for mining models integration turns the mining model into a context model with enhanced semantic meaning of its rules, initially extracted from text by machine learning algorithm.

The framework presented has been focused primarily on the terminological part of the process of ontology building. Future work is intended in enhancing its axiom part as well as on mapping of other text mining models on ontology.

REFERENCES

Albitar, S., Fournier, S., Espinasse, B., 2012. The Impact of conceptualization on text classification. In X. Wang, I. Cruz, A. Delis and G. Huang (Eds.), *Web Information Systems Engineering – WISE 2012*, Lecture Notes in Computer Science, Vol.7651, pp.326-339, Springer-Verlag Berlin Heidelberg.

Baader, F., Horrocks, I., Sattler, U., 2007. Description logics. In *Handbook of Knowledge Representation*. Elsevier.

Bloehdorn, S., Cimiano, P., Hotho, A., Staab, S, 2005. An ontology-based framework for text mining. In *LDV Forum 20(1):87-112*.

Bloehdorn, S., Hotho, A., 2009. Ontologies for machine learning. In S.Staab and R.Studer (Eds.), *Handbook on Ontologies*, International Handbooks on Information Systems, pp.637-661, Springer-Verlag Berlin Heidelberg.

Bratus, S., Rumshisky, A., Magar, R., Thompson, P., 2009. Using domain knowledge for ontology-guided entity extraction from noisy, unstructured text data, In *Proceedings of the Third Workshop on Analytics for Noisy Unstructured Text Data*, pp.101-106, ACM, New York, NY, USA.

Canadas, J., Palma, J., Tunes, S., 2009. InSCo-Gen: A MDD tool for web rule-based applications, In M. Gaedke, M. Grossniklaus, O. Diaz, (Eds.), ICWE, Lecture Notes in Computer Science, Vol.5648, pp.523-526, Springer.

Cunningham, H., Maynard, D., Bontcheva, K., 2011. *Text processing with GATE (Version6)*, University of Sheffield Department of Computer Science.

Damljanovic, D., Amardeilh, F., Bontcheva, K., 2009. CA manager framework: creating customized workflows for ontology population and semantic annotation. In *Proceedings of the Fifth International Conference on Knowledge Capture*, pp.177-178.

Deliyska, B., Rozeva, A., Malamov, D., 2012. Ontology building by dictionary database mining, In *Proceedings of the 38th International Conference Applications of Mathematics in Engineering and Economics (AMEE'12), AIP Conference proceedings*, Vol. 1497(1), pp.387-394

Elsayed, A., El-Beltagy, S., Rafea, M., Hegazy, O., 2007. Applying data mining for ontology building, In *Proceedings of ISSR.*

Garla, V., Brandt, C., 2012. Ontology-guided feature engineering for clinical text classification, *Journal of Biomedical Informatics*, Vol. 45(5), pp.992-998.

Golbreich, C., 2004. Combining rule and ontology reasoners for the semantic web, In G. Antoniou, H. Boley,(Eds.), *Rules and Markup Languages for the Semantic Web*, pp.6-22.

Horrocks, I., 2008. Ontologies and the semantic web. *Communications of the ACM*, 51(12): 58-67.

Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., Dean, M., 2004. SWRL: A Semantic Web Rule Language Combining OWL and RuleML, W3C Member Submission 21 May 2004, http://www.w3.org/Submission/SWRL/.

Maedche, A., Staab, S., 2000. Mining ontologies from text, In R. Dieng, O. Corby (Eds.), *EKAW 2000, Lecture Notes in Artificial Intelligence*, Vol.1937, pp.189-2002.

Microsoft SQL Server, 2013. SQL Server DevCenter, http://www.msdn.microsoft.com/en-us/sqlserver/default.

Morneau, M., Mineau, G., 2008. Employing a domain specific ontology to perform semantic search. In P. Eklund and O. Haemmerle (Eds.), *Conceptual Structures: Knowledge Visualization and Reasoning*, Lecture Notes in Computer Science, Vol.5113, pp.242-254, Springer-Verlag Berlin Heidelberg.

Motik, B., Patel-Schneider, P.F., Parsia, B., 2012. OWL2 Web Ontology Language Document Overview (Second Edition), W3C Recommendation 11 December 2012, http://www.w3.org/TR/owl2-guide/.

Pan, J., 2007. A flexible ontology reasoning architecture for the semantic web, *IEEE Transactions on Knowledge and Data Engineering*, Vol.19 (2), pp.246-260.

Pellet Reasoner Plug-in for Protégé 4, 2013. Clark & Parsia, http://www.clarkparsia.com/pellet/protege.

Pothipruk, P., Governatori, G., 2005. A formal ontology reasoning with individual optimization: a realization of the semantic web, In M. Kitsuregawa et al.(Eds.), *Web Information Systems Engineering (WISE 2005)*, Lecture Notes in Computer Science, Vol.3806, pp.119-132, Springer.

Protégé 4 User Documentation, 2013. http://www.protegewiki.stanford.edu/wiki/Protege4UserDocs.

Rozeva, A., 2011a., Approach for mining text databases, In *Proceedings of the III International Conference "E-governance"*, Sozopol, Bulgaria, pp.82-87.

Rozeva, A., 2011b., Mining model for unstructured data, In *Proceedings of the sixth International Conference "Computer Science'11"*, Ohrid, Macedonia, pp.411-415.

Rozeva, A., 2012. Application of ontologies for knowledge generation. *Proceedings of the IV International Conference "E-governance"*, Sozopol, Bulgaria, pp. 162-170.

Spasic, I., Ananiadou, S., McNaught, J., Kumar, A., 2005. Text mining and ontologies in biomedicine: Making sense of raw text. *Briefings in Bioinformatics*, Vol.6 (3), pp.239-251, Henry Steward Publications 1467-5463.

Wang, T., Maynard, D., Peters, W., Bontcheva, K., Cunningham, H., 2005. Extracting a domain ontology from linguistic resource based on relatedness measurements. In *IEEE/WIC/ACM International conference on Web Intelligence (WI'05)*, pp.345-351.

Wang, X.H., Gu, T., Zhang, D.Q., Pung, H.K., 2004. Ontology based context modeling and reasoning using OWL. In *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, pp.18-22.

Witte, R., Li, Q., Zhang, Y., Rilling, J., 2007. Ontological text mining of software documents. In Z. Kedad, N. Lammari, E. Metais, F. Meziane, Y. Rezgui (Eds.), *NLDB 2007*, Lecture Notes in Computer Science, Vol.4592, pp.168-180, Springer-Verlag Berlin Heidelberg.

Witte, R., Kappler, T., Baker, C., 2007. Ontology design for biomedical text mining. In C. Baker, K. Cheung (Eds.), *Semantic Web: Revolutionizing Knowledge Discovery in the Life Sciences*, Vol.13, pp.281-313, Springer.

Yankova, M., Saggion, H., Cunningham, H., 2008. Adopting ontologies for multisource identity resolution, In A. Duke, M. Hepp, K. Bontcheva, M.B. Villain (Eds.), *OBI Vol.308 of ACM International Conference Proceeding Series*, pp.6-15, ACM.