Document analysis based on multidimensional ontology of electronic documents

Viacheslav Lanin

***Abstract*—** The paper describes an approach to semantic indexing indexing of electronic documents based on ontology that describes the structure, type of document and its contents. In addition, existing ontology descriptions of documents are considered and the differences between the proposed multidimensional ontology from them are described. The solution of the problem of analysis of administrative regulations is described as an application of the approach. An algorithm for implementing semantic indexing based on multi-agent paradigm is proposed.

***Keywords*—** multidimensional ontology, semantic indexing, intellectual agents.

I. Introduction

Transition from processing structured data to unstructured data processing is observed in modern information systems. New classes of systems, such as social networking, corporate portals, wiki-resources, etc. became an integral part of the information process. The key point of such systems is "content", which concept can be generalized to "electronic document." Unstructured nature of information raises the question of the transition from traditional indexing documents based on unrelated keywords («bag of words») to the so-called semantic (conceptual) indexing. Semantic (conceptual) document indexing is an indexing, in which synonyms are reduced to the same concept, and disambiguation are separated into different conceptual units [3].

Semantic index of document can become the basis for solving many problems in the processing of electronic documents, in particular, their search, analysis and classification, cataloging and efficient storage, generation and support their life cycle. It's needed to have consolidated knowledge about their structure and content.

Base of semantic index is ontological resource in that following information about the following aspects of electronic documents is needed: electronic document format; type of electronic document; the structure of an electronic document.

When ontological resource is created, it includes concepts related to all three aforementioned aspects of a document information representation. Each of them is described by ontology. Concepts of the various aspects have to be linked. Thus, a single ontology of electronic documents is being created. In addition, the resource should support the ability to expand and specify the settings on the solution of specific problems arising in the processing of documents in a variety of information systems throughout their life cycle.

Thus, in the paper existing ontology resources for describing documents will be examined, an approach to the description of multidimensional ontology will be proposed and an algorithm for semantic indexing based on multi-agent approach will be provided.

The reported study was supported by RFBR, research project No. 14-07-31273.

Viacheslav Lanin is with the National Research University Higher School of Economics, Perm, Russia (e-mail: vlanin@live.com).

II. Related Works

A. Existing Document Ontology

Dublin core [4] – is a set of metadata used to describe documents of various types (publications, audio records, video records). This set specification has status of official international standard (ISO: 15836 2003). The standard has two levels: Simple, comprising 15 elements and Qualified having three additional elements and element refinements (or qualifiers), which refine semantics of the elements. The main feature of Dublin Core is that every element is optional and might be repeated. Dublin Core is a powerful instrument used to describe resources of various types. The fact that it is widespread and flexible is its overwhelming advantage. However, it describes documents tags, i.e. information having indirect correlation with the document content. In this case it is impossible to describe other aspects of the electronic document.

Project ontologies «docOnto» [3] developed by German research group KWARC (Knowledge Adaptation and Reasoning for Content) differ from other projects oriented on formal structure description development (CNXML document ontology) and document semantics (OMDoc document ontology). Members of this group also develop mechanisms of semantic document indexing and tools for document processing. CNXML document ontology (Connexions Markup Language) describes such terms as paragraph, section, reference etc. Ontology is formalized on UML. It gives detailed description of the document. Unfortunately, work in this direction is frozen, last changes date back 2007. One more direction in document ontologies creation is semantics description of documents for narrow subjects, where documents are well formalized, for example mathematical

ISBN: 978-1-61804-237-8

OMDoc documents. Mathematical Terms, theorems and several other terms are included in ontology.

Document ontology SHOE [5] describes most types of documents. Academic papers are given particular emphasis. Dublin Core reference books and Document Classifier PubMed were the resource.

Document Ontology of Research Centre Linked Data DERI is developed by scholars of Irish Institute DERI (Digital Enterprise Research Institute) and is described in RDFS and OWL-DL [9]. Terms referring project activity documentation are given I the ontology. Developers purposefully refused modelling structure and document content to accommodate flexibility and interoperability.

Muninn project document ontology became the result of processing archive documents of the First World War within the project Muninn WW1 [7]. The Ontology describes bibliography, origins and storage description of the digital item. Most ontology classes are child classes of FOAF. That decision was compatibility possible, on the other hand, make adding additional features of document processing possible, i.e. features for representation document pages, copyright description, etc. One of the main ontology classes is Document, which is integrate class of FOAF Document and Creative Commons Works. Page class describes document pages, in its turn, Image class describes digital page image. Description of different document aspects, document structure in particular, is a significant benefit of this ontology. However, structure description is initially oriented on digital images of archive documents.

Each listed above document ontology has its advantages and disadvantages. We create own ontology specialized on academic paper description.

B. System for create text-based ontology

Nowadays there are some information systems that let you create text-based ontology models of documents or let you define correspondence of ontology models thereby transform one model onto another one. We found two web-resources that let you create ontologies: OwlExporter and OntoGrid.

The core idea of OwlExporter is to take the annotations generated by an NLP pipeline and provide for a simple means of establishing a mapping between NLP (Natural Language Processing) and domain annotations on one hand and the concepts and relations of an existing NLP and domain-specific ontology on the other hand. The former can then be automatically exported to the ontology in form of individuals and the latter as data type or object properties [14].

The resulting, populated ontology can then be used within any ontology-enabled tool for further querying, reasoning, visualization, or other processing.

OntoGrid is an instrumental system for automation of creating domain ontology using Grid-technologies and text analysis in natural language.

This system has bilingual linguistic processor for retrieving data from text in natural language. Worth D. derivational dictionary is used as a base for morphological analysis [13]. It contains more than 3.2 million word forms. The index-linking process consists of 200 rules. “Key dictionary” is determined by words allocation analysis in text. The developers came up with new approach of revealing super phrase unities that consist of specific lexical units. The building of semantic net is carried out this way: the text is analyzed using text analysis system, semantic Q-nets are used as formal description of text meaning. The linguistic knowledge base of text analysis system is set of simple and complex word-groups of the domain. This base can be divided into simple-relation-realization base and critical-fragment-set, that let you determine which ontology elements are considered in this text. The next step is to create and develop the ontology in the context of GRID-net. A well-known OWL-standard is used to draw the ontology structure.

III. USING SAMPLE

Consider the example of the proposed approach based on the work with electronic administrative regulations (EAR) [9]. The basic approach to the development of software tools to support the EAR conduct is ontological modeling. Used in the process ontologies are placed in multi-level repository [10], which contains the domain ontology and ontology normative-reference documents. Domain ontology defines the terms used in the documents, namely it describes concepts such as "process", "operation", "artist", etc., in addition, there are included the various classifiers. Ontology of normative-reference documents, in particular, the ontology of the regulation describe the structure of the characteristic elements of documents.

As a result of text description analysis (decomposition) will be built a conceptual model of regulation that, first, to allow it to verify (check structure, identify duplication of information, etc.), and secondly, will link the fragments of a text document with the relevant concepts of the ontology. In addition, the conceptual model of documents could be used to set the "semantic" relationships between different documents and visualization of these links.

Next, consider how ontologies are used in multi-agent semantic indexing algorithm. Domain ontologies used at semantic analysis step. Ontologies that describe the structure of the document (for example, the aforementioned ontology of regulatory-reference documents) are used at the stage of structural analysis. All ontological resources described in RDF-format. Consider in more detail the steps of the analysis of documents used in the algorithm based on semantic indexing agents.

IV. MULTI-AGENT SEMANTIC INDEXING ALGORITHM

Simplifying the problem we assume that first step of text analysis process was made (for instance using Yandex Mystem[11]), i.e. a set of morphological descriptors for each word have been obtained. All others steps are performed by agent-based semantic indexing. As it could be seen on fig. 1

syntax analysis is not used because it has high time complexity. Instead of this words order in sentence is considered.

Next step is a semantic analysis. The result of the semantic analysis is a semantic descriptor of plain text that binds the morphological descriptors to the elements of the domain ontology. Stop words are skipped.

Next step is a structural analysis. The structural analysis uses document's structure, ontology that describes structure and semantic descriptors of plain text. At this step every concept of structural ontology tries to binds to corresponding structural document element. The result of structural analysis is semantic descriptor of whole text.

Descriptors (morphological, semantic) are a set of tags, which marks each words in the text.

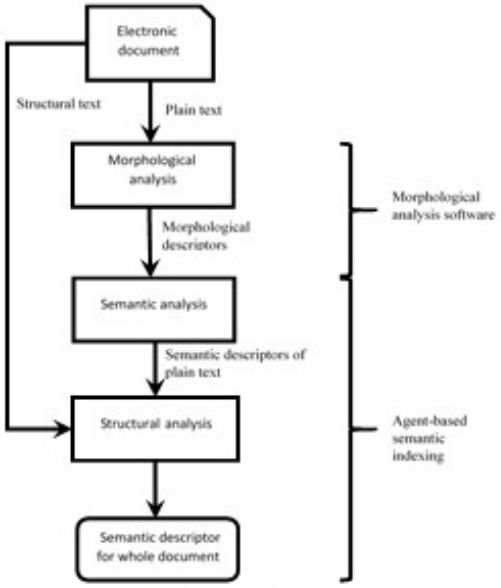

Fig. 1. Steps of document analyses

A. Agent-based solution

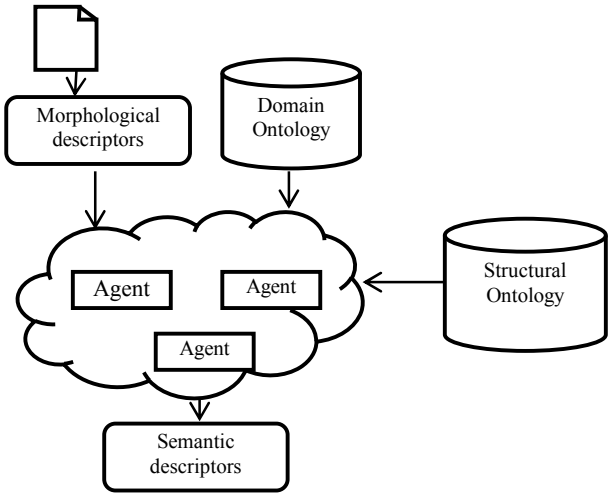

Fig. 2. Steps of solution

Further let us consider the process of building a semantic index based on multi-agent approach (see Fig. 2).

Agents have access to a domain ontology, structural ontology, morphological descriptors and electronic documents which will be indexed. Indexing process is produced on the sentences in the text. Sentences are processed sequentially by agents. The agents form a "team" to index the particular sentence. Thus, agents in the system after the start of the indexing are divided into teams.

B. Agent Types

The following types of agents are identified in the system, according to the functional separation:

- Team Lead First Level Agent - TLFL agent,
- Team Lead Second Level Agent - TLSL agent,
- Word Indexer Agent - WI agent,
- Index Writer Agent - IW agent.

The task of WI agent is accessing to the domain ontology and obtaining the set of possible semantic tags for the indexed word. An input word is passed to the WI agent for indexing with the parameters obtained at the stage of morphological analysis. Resulting set of possible semantic tags is passed to the TLSL agent.

TLSL agent binds to morphological descriptors of the sentence and distributes words to all available WI agents. TLSL agent finishes its work on the sentence when the consistent semantic descriptor is formed and written to the document. TLSL agent plans actions for the WI agents and participates in the auction for the resolution of contradictions. After building a consistent semantic descriptor TLSL agent transmits the generated semantic descriptor of the sentence to IW agent who writes semantic tags to the document.

TLFL agent binds to morphological descriptors of the document and distributes descriptors of the sentences to all available TLSL agents. TLFL agent monitors the work of TLSL agents. If the work on the sentence is completed TLSL agent gives TLFL agent a new sentence. In addition, TLFL agent conducts an auction among TLSL agents to resolve ambiguity in the descriptors (see details in section «Agent negotiation»). Besides TLSL agents perform structural analysis. They distribute parts of structural ontology to TLSL agents.

C. Agent communication

Agents communicate through language FIPA ACL (Agent Communication Language developed by FIPA) [8]. Two types of actions are used. They are inform (inform about anything) and perform (execution of an action).

Inform action type is implemented in the following cases:

WI agent informs the TLSL agent of completion of indexing word and give it the set of possible semantic tags; content of the communication is as follows: (id, tags), where the id is the identifier word that came to be indexed, tags are returned set of possible semantic tags;

TLSL agent informs the TLFL agent of completion of

ISBN: 978-1-61804-237-8

indexing sentence with a specific identifier; content of this message contains an identifier of indexed sentence.
Perform action type is implemented in the following cases:

TLFL agent gives to the TLSL agent a task to index a sentence with a specific descriptor; content will look like this: (id, descriptor), where the id is the identifier of the sentence, descriptor is descriptor of the sentence received as a result of syntactic and semantic analysis;

TLSL agent gives a task to the WI agent to index a word with specific id; content will look like this: (id, word, parameters), where id is ID of the word, word is the word for indexing, parameters are parameters obtained at the stage of morphological and syntactic analysis;
TLSL agent gives a task to the IW agent to write semantic tag of specific word; content is as follows: (word, tag), where the word is an indexed word, tag is just a semantic tag of indexed word.

D. Planning

The planning is dynamic. TLSL agents themselves form a team of agents from the available WI agents. A count of needed WI agents depends on structure of a sentence. With a lack of WI agents at the time of formation of the team TLSL agent may designate to perform indexing of few words at once to the same WI agent. TLFL agent monitors the performance of work of TLSL agents and if they are released it assigns them new sentences for indexing. Completing of work of the agents (WI and TLSL) monitored not only by sending their corresponding messages of inform type, but also change their states (agent states) in the meaning of "vacant."

E. Agent knowledge bases

WI agents and IW agents are primitive reflex agents working in the mode of stimulus-response. Their main function is a simple, no inference, execution of work. In the knowledge bases of these agents are only procedural steps.

Knowledge bases of TLFL and TLSL agents represent productions with embedded procedural actions. In fact, the script actions are necessary for the distribution of work between agents. Accordingly TLSL agent knowledge base contains a script for word distribution among WI agents, and TLFL agent knowledge base includes a script for sentences distribution between agents TLSL.

F. Agent negotiation

TLFL agent conducts an auction among agents TLSL, each of which has a contextual memory (training component). Every TLSL agent using the contextual memory votes for a one option of sematic descriptor of the sentence. Option of semantic descriptor of the sentence with the highest number of votes will be considered as a true semantic descriptor of the sentence. The set of all consistent semantic descriptors of the sentences form the document semantic descriptor.

V. CONCLUSION

Unlike existing ontologies describing documents multidimensional ontology represents the document structure, which allows to consider this information during indexing process. In developing ontologies it included the mechanisms for integration with domain ontologies and expanding of ontology - adding new "aspects", which also expands the scope of the decision. The proposed multiagent approach creates preconditions for solving the optimization problem of parallel execution of semantic indexing.

Also planned that the developed ontology and algorithms will be used in a number of projects related to the development of domain-specific languages (Domain Specific Languages, DSL) for different domains based on linguistic tools MetaLanguage.

REFERENCES

[1] Segaran T., Evans C., Taylor J. Programming the Semantic Web, O'Reilly Media, 2009.

[2] Lukashevich N.V., Dobrov B.V. Bilingual information retrieval based on the automatic conceptual indexing // Computational linguistics and intelligent technologies. Proceedings of the International Conference “Dialogue-2003”. Protvino. June 11-16 2003y. / Ed. by I.M.Kobozevoy, N.I.Laufer, V.P.Selegeya - M.: Science, 2003. - pp.425-432.

[3] CNXML/DocumentOntology http://mathweb.org/wiki/CNXML/DocumentOntology

[4] Dublin Core Metadata Element Set, Version 1.1 http://dublincore.org/documents/dces/

[5] Document Ontology (draft) http://www.cs.umd.edu/projects/plus/SHOE/ onts/docmnt1.0.html

[6] Grishman. R. TIPSTER Architecture Design Document Version 2.3. Technical report, DARPA, 1997. http://www.itl.nist.gov/div894/894.02/related_projects/tipster/.

[7] Muninn Documents Ontology http://rdf.muninn-project.org/ontologies/documents.html

[8] XML Languages http://cnx.org/help/authoring/xml

[9] Lanin VV Lyadova LN Technology of support maintenance of electronic administrative regulations based on ontological models / / Proceedings of the All-Russian conference with international participation "Knowledge-Ontology-Theory." Novosibirsk, 2011, pp. 38-46 V.2.

[10] Lanin V.V. Using multi-level ontology repository for electronic document analysis / / Proceedings of international scientific conference "Intelligent systems» (AIS'08) and "Intelligent CAD» (CAD-2008). Scientific publication in 4 vols. T. 1. - Moscow: Fizmatlit 2008. Pp. 202-206.

[11] Program for morphological analysis of text in Russian “Mystem”. [Electronic resource] [Mode of access:http://company.yandex.ru/technologies/mystem/] [Checked at: 24.06.12]

[12] D. Worth, A. Kozak, D. Johnson “Russian Derivational Dictionary”, New York, NY: American Elsevier Publishing Company Inc, 1970

[13] R. Witte, N. Khamis, and J. Rilling, “Flexible Ontology Population from Text: The OwlExporter” Dept. of Comp. Science and Software Eng. Concordia University, Montreal, Canada. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2010/pdf/932_Paper.pdf