One approach to document semantic indexing based on multi-agent paradigm

George Sokolov
Computer Science Department
Perm State University
Perm, Russia
sokolovgeorge@gmail.com

Viacheslav Lanin
Department of Business Information Technologies
National Research University Higher School of Economics
Perm, Russia
lanin@perm.ru

The problem of search pertinence increasing with a low time-complexity is one of the major research issues in Computer Science. Semantic search as an alternative solution to this problem has a high time complexity. This paper describes the use of agent-based approach to reduce the time complexity of constructing semantic indexes used for searching.

Semantic indexing; agent; ontology; document

I. INTRODUCTION

Nowadays the information retrieval (from the Internet and off-line sources) is one of the major research areas in Computer Science. The main criteria of a successful search are the high relevance of search query information and fast response time. Traditional search engines typically use an approach «Bag of words» based on statistical methods to search for information. This approach takes precedence over semantic search methods is due to low time-complexity, low implementation complexity and satisfactory degree of relevance. One of the main areas of modern researches in the information retrieval is an increasing of search pertinence with a low time-complexity.

In syntactic search some indexes are built to find quickly the information required on some key words. By analogy let's introduce a concept of a semantic index. In this paper the semantic index is one-one correspondence between elements of the text and concepts from some ontological resource. There are different formats of the semantic indexes. Some of them are primitive (such as microformats hCard, Geo, microdata html5) and other formats are advanced (such as RDF, OWL, underlying the Semantic Web). In the semantic indexing there are two directions: the construction of semantic indexes and search for information on a semantic index. In this paper we will consider the construction of the semantic index (or the so-called semantic markup) for electronic documents.

The main problems of constructing semantic indexes are

1) high time-complexity (is due to various kinds of ambiguity that require paying respect of a context),

2) the problem of choosing ontology, which would be sufficiently complete to satisfy all search queries in an electronic document,

3) large amount of constructed semantic indexes and the problem of storage.

In this paper, the authors offer one approach of solving the first problem (the problem of time-complexity). Obviously, increase in the rate of the semantic indexing operation is required not one but several calculators, i.e. the parallelization of this operation is needed. The execution of the semantic markup operation requires the coordination of actions to resolve ambiguities. That's why simple asynchronous calculators aren't capable to solve the problem. According to the authors the most appropriate solution is using agent-based approach.

II. EXISTING APPROACHES

Solution to the agent-based semantic indexing problem can be obtained in two ways:

1) using of generic agent-based platforms that can decide a wide range of tasks,

2) using of specialized semantic indexing systems based on the multi-agent paradigm.

Let us consider each of these methods. Most popular agent platforms are JADE [1], MASDK [2], Zeus [3].

TABLE I. GENERIC AGENT-BASED PLATFORMS

	JADE	**MASDK**	**ZEUS**
Developer community	Telecom Italia Lab	SPIIRAS	BT Laboratories
License	LGPL	LGPL	LGPL
Description	This is the platform for rapid development of multi-agent systems, which implements FIPA standards [4]. JADE provides base classes for creating agents and infrastructure for the operation of multi-agent system.	This is the software environment for multi-agent application development that supports the full life cycle application development of MAS. The agent platform, which is the part of MASDK, works on the principle of P2P.	This is the agent platform designed for rapid development of multi-agent applications. Zeus provides a library of agent components.
Description of the agent behavior	Set in the code of the agent class that inherits from Agent.	Set with language ASML. This language is used for generating applied MAS.	Set in an environment for building agents, from which the agent code is generated.

Each of these agent platforms allows one way or another to describe the behavior of the agent. Depending on the platform we can define almost any behavior of an agent, programming or describing it using specific language. So we can determine the behavior of the agent that implements mechanisms of semantic indexing. The key problem of this approach is the high overhead of run-time. This is due to a complex infrastructure applications received applications. This can be compared with a programming in high level language and Assembler. The actions are the same, but the performance is significantly different. Therefore, such an approach to the problem is not satisfactory.

As noted above, the second approach to the problem of semantic indexing is the use of specialized semantic indexing systems based on the multi-agent paradigm. In this area, it was found only one solution – Magenta Toolkit [5]. This software solution is commercial, so there is no legal possibility to evaluate the effectiveness of work and, especially, to study the mechanisms of their internal functioning. Magenta Toolkit developers have written a number of publications [6, 7], which describe the principles of the system in outline without specifics. This decision is also not satisfactory.

Therefore, the task of the research is development of an open (open source and detailed descriptions of the principles) and an effective method of semantic indexing based on the multi-agents paradigm. In addition, you also need the option to apply this method to all electronic records. So the agent platform must be developed.

III. DOCUMENT ANALYSES STEPS

On Fig. 1 text mining process steps are shown. Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar.

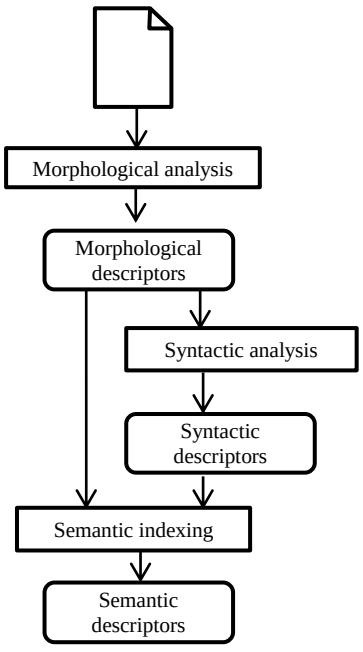

Figure 1. Steps of document analyses

Simplifying the problem we assume that first two steps of text mining process have been made, i.e. a set of syntactic and morphological descriptors for each sentence have been obtained. The result of semantic analysis (indexing) is a semantic descriptor of text that binds the syntactic descriptors of sentences to the elements of the domain ontology which is used for semantic search.

Descriptors (morphological, syntactic, and semantic) are a set of tags which marks words in the sentence. Syntactic and morphological descriptors will be put into relational tables for two reasons. Firstly, syntactic and morphological descriptors will be actively used for semantic indexing. Secondly, we don't want to pile up document by tags. Each word in the text (except for a different kind of stop words) will be assigned a unique identifier. Each identifier corresponds to a separate table row.

Thus, *i*-th row of the table looks like $(id_i, \{a_j\}_i)$, where id_i – the identifier of the word, $\{a_j\}_i$ – set of attributes (tags) that have been assigned to a given word during morphological and syntactic analysis process. In each row of syntactic descriptor table an identifier of applicable syntactic rule is indicated. The syntactic rule is a rule for constructing syntactically correct sentences. The semantic descriptor is represented as set of tags (semantic markup) within the indexed document.

IV. AGENT-BASED SOLUTION

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

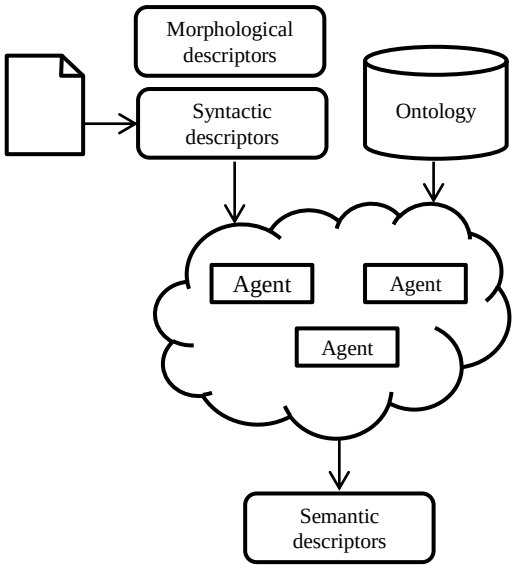

Figure 2. Arhitecture of agent platform

Agents have access to a domain ontology, syntactic, 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.

A. Agent Types

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

1) Team Lead First Level Agent - TLFL agent,
2) Team Lead Second Level Agent - TLSL agent,
3) Word Indexer Agent - WI agent,
4) 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 and syntactic analysis. Resulting set of possible semantic tags is passed to the TLSL agent.

TLSL agent binds to syntactic and 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 also 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 syntactic, 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»).

B. 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:

1) 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;
2) TLSL agent informs the TLFL agent of completion of 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:

1) 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;
2) 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;
3) 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.

C. 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."

D. 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.

E. 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 shall 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

So, in this paper we have discussed various approaches to solving the problem of document semantic indexing based on multi-agent paradigm. We propose a variant of the solution of that problem and describe it in terms of morphological, syntactic and semantic descriptors of the text. Specialized types of agents are introduced and the general principles of multi-agent system functioning are described.

References

[1] JADE Programmers guide. http://sharon.cselt.it/projects/jade /doc/programmersguide.pdf

[2] Gorodetsky V., Karsan O., Samoilov V., Serebryakov S. "Applied multi-agent systems of group control" / / Artificial intelligence and decision making № 2.2009

[3] ZEUS Techical Manual. www.upv.es/sma/plataformas/zeus/Zeus-TechManual.pdf

[4] The Foundation for Intelligent Physical Agents. http://www.fipa.org

[5] The Magenta Toolkit. http://www.magenta-technology.ru/ru/

[6] Andreev V., Iwkushkin K., Karyagin D., Minakov I., Rzevski G., Skobelev, P., Tomin M.: Development of the Multi-Agent System for Text Understanding. In 3rd International Conference 'Complex Systems: Control and Modelling Problems'. Samara, Russia, September 4-9 2001, 489 – 495.

[7] Minakov I., Rzevski G., Skobelev, P., Kanteev M., Volman S. : Multi-Agent Meta-Search Engine Based on Domain Ontology. http://www.magenta-technology.ru/ru/

[8] FIPA ACL Message Structure Specification. http://www.fipa.org/specs/fipa00061/SC00061G.pdf