JOURNAL OF APPLIED
COMPUTER SCIENCE
Vol. 24 No. 2 (2016), pp. 17-37

Retrieval and Processing of Information with the Use of Multi-Agent System

Aneta Poniszewska-Maranda, Lukasz Gebel

Institute of Information Technology
Lodz University of Technology
Wolczanska 215, 90-924 Lodz, Poland
aneta.poniszewska-maranda@p.lodz.pl

Abstract. *Today the amount of information available in the world is so large that it exceeds significantly the perceptual capacity of a man. The progress of science and increased access to the media, in particular to the Internet, allows an increasing number of people to create and publish their own content. Refer to the information that can be found in our area of interest is limited to read a very small part of the works on the topic. What's more the overlooked information can be very useful, but selective viewing of the contents and limited time resources prevent the users to meet their information needs. An attempt to solve this problem are multi-agent systems, using intelligent methods of retrieval and processing of information.*
Keywords: *intelligent agent, multi-agent system, retrieval of information, intelligent processing of information, sentiment analysis.*

1. Introduction

Information systems are increasingly used to carry out the tasks that were required the human involvement and the use of his natural intelligence. What's more, many of these tasks exceeds the capacity of human perception due to its complexity and the size of the data that must be analysed in order to execute them. Moreover, the process of design and implementation of such systems becomes more

and more difficult, not only due to non-trivial scope of issues solved by them, but also on their complex architecture and the need for their scalability. This is due to the dynamic nature of the environments in which such systems operate. The Internet is a data source with a high diversity and complexity and the information systems must deal with it. To a lesser extent, this also applies to the systems for personal computers, which may also be the source of a large amount of data with non-standardized structure. An attempt to solve such problems is the class of systems that perform intelligent processing and obtaining information in multi-agent way.

For many years, the continuous, very dynamic development of information technology can be observed. Initially, computers and software were used mainly as separated units, so the complexity of the created systems was relatively small. With the appearance of Web network the technology development has been focused on the examination of the possibility of building the software that not only uses the network resources, but also becomes their part and co-founder of the Internet. Currently, the universality of network technologies has led to the creation of a huge number of applications and materials. So a large amount of information require a new approach to how it is processing. An example of this new approach is the agent paradigm.

To obtain this purpose it is necessary to investigate the suitability of the theory of multi-agent systems to create the solution that implement intelligent retrieval and processing of information. The exemplary multi-agent system was proposed and it can carry out an intelligent analysis of the repositories of text documents and obtain the information that is naturally specified using the human intelligence. Evaluation of the effectiveness of proposed solutions and their architecture design process based on operation of agents will reply on the merits of the connection of theory of multi-agent systems and intelligent retrieval and processing of information.

The presented paper is structured as follows: section 2 gives the outline of intelligent agents and multi-agent systems, section 3 presents the intelligent information retrieval and processing methods. Section 4 deals with the idea of using the multi-agent systems in the field of intelligent information retrieval and processing and contains a proposal of multi-agent realization of sentiment analysis of text documents while section 5 describes a project named Textlligent which is the practical implementation of algorithms and presents the evaluation of Textlligent system.

2. Multi-agent systems

One of the most cited definition of an agent is the one given by Wooldridge and Jennings. According to this definition *agent* is a computer system located at a certain *environment*, which can take *autonomous decisions* to *achieve their design goals*. The agent is part of the environment which is able to follow. Events occurring in the environment can have an effect on the agent, but only the agent decides to carry out certain actions. Decisions taken by the agent may have an impact on the environment and change its status, but should be consistent with the objective that tries to achieve the agent [1, 2].

An *intelligent agent* is one that is capable of flexible autonomous actions in order to meet its design objectives: reactivity, pro-activeness and social ability. Intelligent agents are capable of interacting with other agents, are able to perceive their environment and respond to changes that occur in it and they are able to exhibit goal-directed behaviour by taking the initiative. And, of course, all these functions are made by an agent in order to satisfy their design objectives. Agents operate and exist in some environment that typically is both computational and physical. The environment might be open or closed, it might or not contain other agents. At times, the number of agents may be too numerous to deal with individually and it is more convenient to deal with them collectively as a society of agents. An agent has the ability to communicate. This ability is part perception (the receiving of messages) and part action (the sending of messages). Agents communicate in order to achieve better goals for themselves or for the system in which they exist [1, 3, 2].

Multi-agent system is composed of multiple interacting software components known as agents, which are typically capable of cooperating to solve the problems that are beyond the abilities of any individual member. A multi-agent system consists of a number of agents that interact with one another. In the most general case, the agents will be acting on behalf of the users with different goals and motivations. To successfully interact, they will require the ability to cooperate, coordinate and negotiate with each other, much as people do [1, 3, 4, 5].

Multi-agent system is an information system which consists of many elements called agents which work in environment or environments. Agent is a computer program that can autonomously perform actions to meet it's design objectives. General schema of such systems is presented in figure 1 [1, 4, 6].

Another definition describes multi-agent systems as computational systems in which agents work together to achieve common goal. Thanks to the cooperative

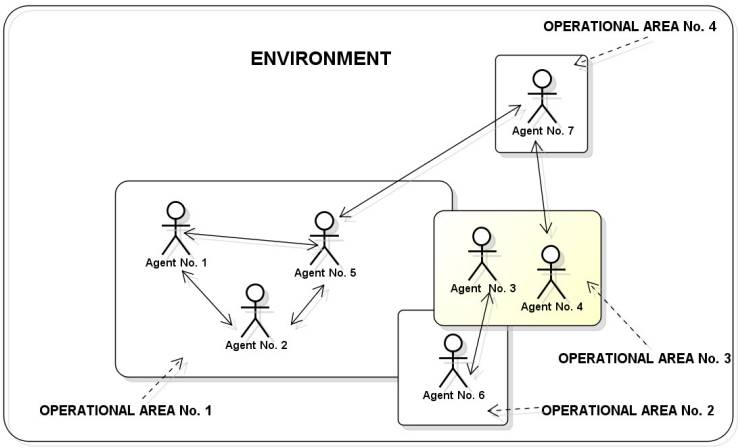

Figure 1: General schema of multi-agent system

approach of agents the work such systems are more effective then classical, centralized systems [7, 8, 9, 10].

3. Intelligent retrieval and processing of information

Development of science and increased access to media, in particular to the Internet makes it easy to create and publish own content. Everyone can share their creativity in the form of article, book, photos or paintings. Finding and reading all of interesting materials is probably impossible but we can omit important sources of information.

Information retrieval and processing methods can be helpful to cope with such problems. This computer science domain is defined as searching for information in documents which do not have unified internal structure. Such documents are usually text documents which are written in natural language. It makes it harder to extract required information automatically, because usually natural, human intelligence is needed in such tasks. What is more such exploration of text document can be very computationally complex which creates the need for effective solutions [11, 12].

Information retrieval combined with its appropriate processing and analysis, leading to the discovery of hidden patterns and regularities. It can be said that it

is a part of a larger field called data mining, dealing with automatically discovering all kinds of relationships in large data sets. Documents written in a natural language are characterized by a large variety of ways in which their content is formulated and saved. This is due to the multitude of languages which are present in the world and their more or less flexible rules and a way of expressing the content by individuals. For these reasons, we need the methods that allow the processing of documents analysed in such a way as to meet the accepted unified rules. The initial processing of the documents makes their representation unambiguous during the further stages of information retrieval.

One of the first stages of the initially processing of text documents is the separation of the individual words. There are three main types of units in word tokenization, named also word atomization: tokens, types and terms. The words stemization is the process of extraction of core words. It usually means the removal of prefixes and postfixes existing in word or other word's elements that do not belongs to the core. The process of stemization is usually closely related to the language in which the saved document is processed.

The classification process of set of documents is composed of the following stages:

- collection of documents, through which it will be possible to generate the useful classifier,
- preliminary processing of text documents, containing the stemization and tokenization of words in a sentence,
- initially processed documents are divided into test set and training set – training set is used to learn a classifier and test set to validate its effectiveness,
- classifier training with the use of extracted features or whole representations of objects – the training typically involves the presentation of examples in the form of vectors of features or objects and giving to which class they should belong to,
- testing of classifier's results on the documents coming from the test set – if the classifier has acquired in the course of training the generalizing properties then it should effectively classified the documents, which were not used for a training.

A very important element of the classification is the extraction of features. The extraction of features in text documents is mainly for finding various dependencies between the words of these documents. This is a very intuitive approach, because we can expect that certain sets of words mainly occur in the texts coming from specified domain. The best selection of features should allow as much as possible the degree of separation documents belonging to different classes. If the values of the selected features are not significantly different for documents with different labels, this prevents the correct classification.

One of the most commonly used methods of feature extraction is the use of *term frequency method (TF)*. The frequency in the case of information retrieval means the number of occurrences of the word in the document. There are some approaches to the formulation of the frequency of terms in text documents. The simplest of them, shown in formula 1 involves counting the number of occurrences of the word t in the document d [11].

$$TF(d,t) = f(d,t) \tag{1}$$

Another useful method is use of *Inverse Document Frequency (IDF)*. Unlike TF measure which treats every word equally, the IDF is built with intuition that words which occur very often in many documents are less informative then words which occur often in specific documents. IDF measure is defined in formula 2. This measure does not apply to one document, but it states the weight of validity of a term in the entire collection of documents. $|d|$ is a number of all documents in a given documents set, $|d_t|$ is a number of documents which contain the word t. The bigger is the value of $|d_t|$ the smaller is the IDF measure [11, 12].

$$IDF(t) = log\frac{1+|d|}{|d_t|} \tag{2}$$

IDF measure is combined with TF measure and named TF-IDF measure. It is defined by formula 3. It means that for every word in a document its TF measure value is multiplied by IDF value of this word for the whole currently analyzed set of documents. With the use of such measures we can build the feature vectors containing the values for each word of a document from the set [11, 12].

$$TF - IDF(d,t) = TF(d,t) \times IDF(t) \tag{3}$$

Described methods of information retrieval and processing allow for its searching be more intelligent and meet the expectations of users, as well as make it easier

to find the complex relationships between the texts. General process of classification and grouping is suitable for the analysis of data of various types, however, the selection of appropriate methods for extracting the features and measures of similarities allows to adjust it to be used for the analysis of text documents.

The problem of information overload, which meets every day the average person is not associated only with the web network. It refers also the information stored in different information systems of the companies. To solve this problem the multi-agent systems which use intelligent methods of information retrieval and processing can be used. Such systems consist of information agents which are the units that have the ability to extract the information from given repositories, for example repository of text documents. Repository can be understood as any set of information like database, website or even the whole Internet. *Information Agents* should have an access to at least one such repository and be able to solve some kind of problem posed by the user and related to this repository [1, 2].

Figure 2 presents the schema of multi-agent system for information retrieval and processing. The user of such system instructs it to perform some task by using the communication module. This module consists of the interface, allowing the option choice to define the problem and will vary depending on the type of application. Information Agents receive the orders from communication module and try to retrieve information from repositories and process it so it will be readable and understandable to the user. Collected information is transferred to the communication module and presented to the user. Information Agents can have access to different repositories which will affect the results of their work. What is more agents can communicate with each other to improve the results.

Multi-agent system with information agents should remind the team of experts who receive an order relating to their area of knowledge (in the case of agents these are the repositories to which they have an access), and provide the customer (system user) response searching by him. Problems solved by agents and detailed system architecture depend on the specific application [1, 2].

4. Multi-agent intelligent algorithms of sentiment analysis basing on content of text documents

One of the information retrieval tasks is an analysis of sentiments contained in the content of text documents. Solution to this problem is an automatic method that will determine if a given document is polarized positively or negatively. The

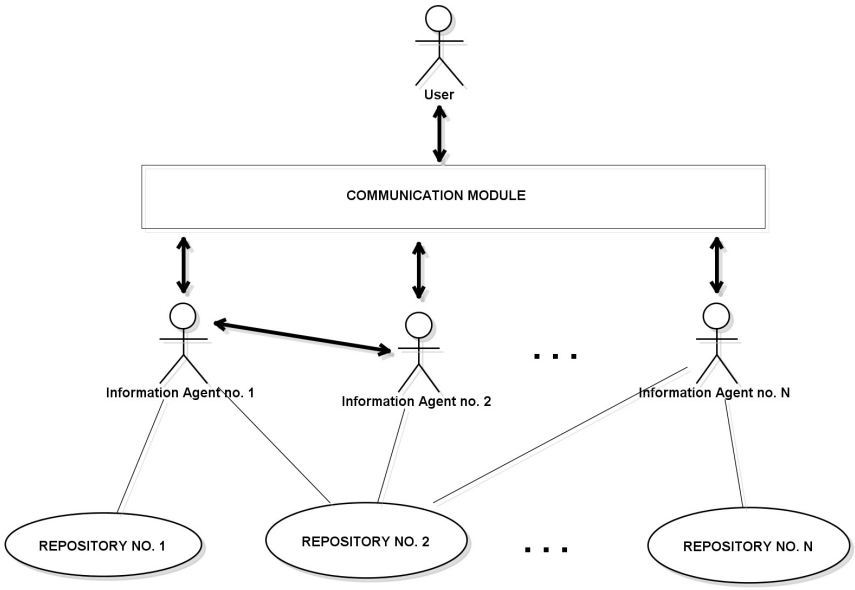

Figure 2: Schema of multi-agent information retrieval and processing system

Internet users very often search for and read the content that contains opinions about a subject of their interests. What is more, nowadays exchanging points of view with the use of the Internet became a trend and is very common [11, 13].

It is easy to imagine a situation, when a person who is trying to make decision about buying some product, would like to know other people's opinions about it. But the problem is a huge number of available opinions, which can be found on the various websites. Reading all of these opinions is practically impossible. Similar problem concerns every domain in which user can find a big amount of opinions expressed in natural language, for instances reviews of movies, books or restaurants. It creates the need for tool, that would be able to analyse the content of such opinions and intelligently determine their positive or negative polarization.

In order to depict how much non-trivial is automation of process of sentiments analysing, let's consider an example of a short movie review:

"The movie was very interesting and full of action. Excellent realization of special effects was unfortunately overshadowed by poor acting."

Above example cannot be clearly classified as positive or negative. For some

people this review will be in general positive, while for other negative parts will be more important rather then positive.

An attempt to solve the problem of automatic detection of polarization of text documents can be the approached based on classifying the feature vectors or lexicons of words. Description of these methods and proposals of their multi-agent realizations can be found in the below subsections.

4.1. Multi-agent sentiment analysis algorithm based on classifying feature vectors

First solution of determining the polarization of text documents is algorithm based on classifying the feature vectors. It consists of typical stages associated with documents classification process: preprocessing, feature extraction and classification of extracted vectors.

Considering large number of analyzed documents and their various sources it seems reasonable to propose the multi-agent realization of such algorithm. Algorithm 1. uses the intelligent agent concept and contains four types of agents:

1. Manager Agent,
2. Preprocessing Agent,
3. Extractor Agent,
4. Classifier Agent.

Manager Agent is a coordinator who manages all steps of algorithm process. He can creates another types of agents which can perform the tasks needed in particular steps of the process. Manager Agent also collects the results of Agents' work and presents it to the user.

Next type of agents are *Preprocessing Agents*. Their job is to load the text files from algorithm's input path and depending on given configuration, performing stemization of words in loaded document and removal of stop words. After preprocessing this agent returns text document to Manager Agent.

Extractor Agents have to extract the features of document and build vector of these features' representation. For the process of learning and testing agents the vectors should be labelled with tags "positive" or "negative".

Last type of agents in proposed multi-agent realization of algorithm are *Classifier Agents*. These agents are trained with the use of previously prepared feature

vectors. After training they are tested to determine the evaluation of their activity. The ability of classification of these agents gives them purely autonomous character, because their decision cannot be forced by no means, but it is an effect of intelligence module which they have, so the trained classifier.

Multi-agent sentiment analysis algorithm is presented in figure 3. This algorithm consists of three main stages. First one is *preprocessing of documents*. Manager Agent creates specified number of Preprocessing Agents. This number can be different depending on hardware resources used for execution of algorithm. Every Preprocessing Agent works parallel with other Agents. Thereby many documents can be preprocessed at the same time. Agent which ended preprocessing of one document loads the next one until all documents will be preprocessed. Except loading document to the computer memory Preprocessing Agents can perform optional stemization and stop words removal (steps 5. and 6. of algorithm 1).

Next step is *extraction of documents features*. This time Manager Agent delegates this task to the Extractors Agents which build the feature vector with the use of chosen extraction methods like for instance word frequency or TF-IDF measure. Extractor Agents work parallel until all of the documents from set of documents have the vector representation.

After getting all of feature vectors the *k-cross validation process* is tun. Its purpose is to train and test the Classifier Agents. This process (Fig. 4) involves on dividing the set of feature vectors on k equal parts in which the number of positive and negative documents are relatively close. Next, every part of the set is used for training of different classifiers being the decision-making core of Classifier Agent, while the rest part of the set is used to test the effectiveness of such agent.

What is important, the algorithm is very universal and allows using any classifier that can be trained with the use of feature vectors and their labels. Effectiveness of this solution can be measured by using the precision rate which is presented in formula 4. Correct response is the situation when agent correctly identified the appropriate class of input document as "positive" or "negative". A sum of all responses is equal to number of all documents in test set [11, 13].

$$precision = \frac{number\ of\ correct\ responses}{number\ of\ all\ responses} \tag{4}$$

At the end of the algorithm Manager Agent computes the average precision of trained Classifier Agents and returns it to the user. When the result is satisfying, one Agent can be trained with the use of bigger number of documents. Otherwise algorithm can be performed once again for another input parameters. Choice of

Algorithm 1 Multi-agent sentiment analysis algorithm for text documents, based on classifying feature vectors

Require: positive and negative documents, parameters of classifiers training

Create Manager Agent
Manager Agent → create N Preprocessing Agents
while not preprocessed all input documents **do**
    Preprocessing Agent → read path and load document
    Preprocessing Agent → [remove stop words]
    Preprocessing Agent → [perform stemization]
    Preprocessing Agent → return preprocessed document
end while
Manager Agent→ create N Extractor Agents
while not extracted features from all preprocessed documents **do**
    Extractor Agent → read preprocessed document
    Extractor Agent → build feature vector for preprocessed document
    Extractor Agent → return feature vector with label "positive" or "negative" depending on class of loaded document
end while
Manager Agent → divide feature vector list for K equal parts
Manager Agent → create K Classifier Agents
for $i = 0; i < K; i++$ **do**
    Train $i - th$ Classifier Agent using $(K - i) - th$ part of feature vectors
    Test $i - th$ Classifier Agent using $i - th$ part of feature vectors
end for
Manager Agent → return average result of classification performed by Classifier Agents

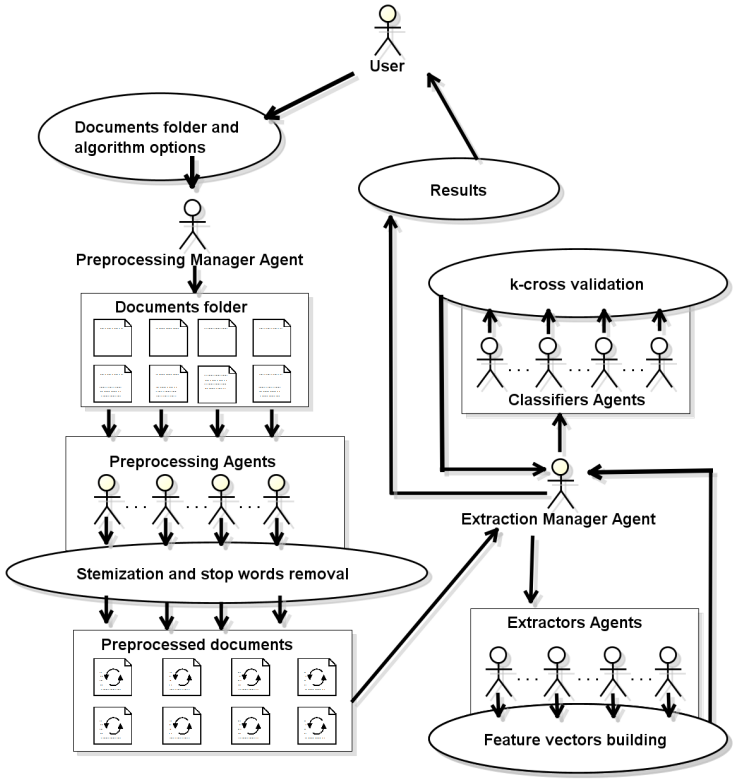

Figure 3: Illustration of multi-agent sentiment analysis algorithm for text documents, based on classifying feature vectors

algorithm's parameters can be done by user of a system or by Manager Agent.

4.2. Multi-agent sentiment analysis algorithm based on lexicon of words

Another way to solve the problem of automatic sentiment analysing based on content of text documents is the use of *lexicons of words*. Such lexicons contain the list of words with appropriate polarization value. It means that every word in lexicon is labelled as "positive", "negative" or sometimes "neutral". There are also much more complex lexicons that assign numerical values to every word. For instance, word "fantastic" could be assigned as 1.0 or 9.0 while word "boring" could be equal to -1.0 or -9.0 [13].

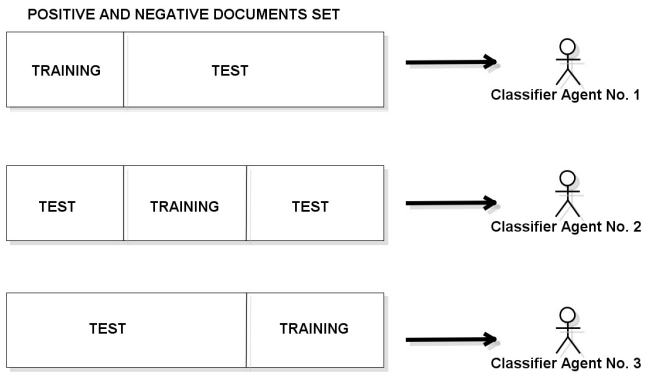

Figure 4: Illustration of k-cross validation process with *Classifier Agents*

What is more besides using these lexicons to classifying documents as positive or negative they can also be used to build user's *own lexicons*. To do this the relationship between words connected with particular connective is used. Some connectives like "and" can be usually found between words with the same polarization, while words connected with "but" are usually differently polarized [13]. Example statements are presented below:

„Interesting and wise"
„Solid, but overrated"

In the first example the word "interesting" is positive so if our lexicon would contain word "interesting" we can add to it the word "wise" with exactly the same polarization. In the second case the word "solid" occurs with word "overrated" and these words are connected with "but". It suggests that second word has different polarization then word "solid", so we can add it to lexicon as negative.

Using the same intuition a multi-agent realization of classification algorithm based on lexicon of words can be used. Such algorithm would be able to extended by given lexicon while working on analysing the sentiments of text documents. Pseudo-code of such process is presented in algorithm 2, while its illustration in figure 5.

The preliminary phase of documents preprocessing is identical as in algorithm 1. New type of Agent proposed in algorithm based on lexicon of words is *Lexicon*

Algorithm 2 Multi-agent sentiment analysis algorithm for text documents, based on lexicon of words

Require: positive and negative documents, basic words lexicon

```
Create Manager Agent
Preprocessing process analogous to algorithm 1.
Manager Agent → create N Lexicon Builder Agents
while not searched in all documents do
    Lexicon Builder Agent → load preprocessed document
    Lexicon Builder Agent → find words connected with "and" or "but"
    if one of the words is not contained in basic lexicon and is connected with
    "and" then
        add word to lexicon with the same polarization as the other word from
        pair
    else if one of the words is not contained in basic lexicon and is connected
    with "but" then
        add word to lexicon with the different polarization as the other word from
        pair
    end if
end while
Manager Agent → create K Classifier Agents
while not classified all documents do
    Classifier Agent → sum up positive and negative words values
    if positive >= negative then
        return response "positive document"
    else
        return response "negative document"
    end if
end while
Manager Agent → return average precision of Classifier Agents
```

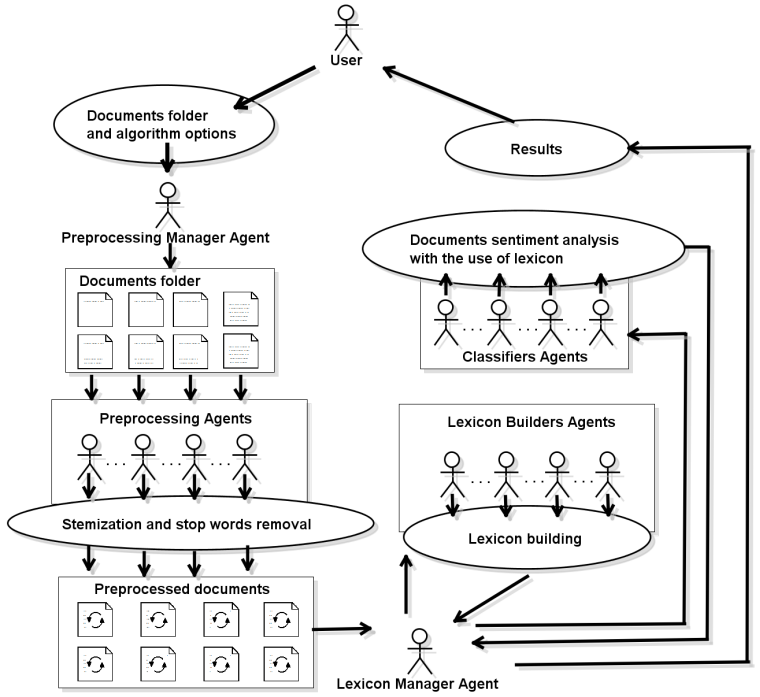

Figure 5: Illustration of multi-agent sentiment analysis algorithm based on lexicon of words

Builder Agent. Agents of this kind search for words connected with appropriate connectives like "and" or "but". These two connectives are just examples and list of useful connectives can be expanded and different for every language. By using particular connectives Lexicon Builder Agents are expanding the lexicon using rules about the same or different polarization.

Classifier Agents work differently then Agents in algorithm based on classifying feature vectors. They use the expanded version of word's lexicon and sum up the values of negatives and positives words in currently processed text document. If the sum of polarization is negative then document is labelled as "negative", otherwise it is labelled as "positive".

It should be noted that Classifier Agents are strictly dependent on lexicon of words. If such lexicon lacks in quality or it is very poor it is expected that the results of agents will not be satisfying. Depending on type of analysed texts it may

be necessary to build the specific lexicons just for them. Words can have different polarization if they occurs in horror book and different in book for children. It can be a key aspect and should be considered while testing the effectiveness of the algorithm.

5. *Textlligent* – multi-agent system for text document analysis

Presented in the previous section the multi-agent realizations of sentiment analysis algorithms can be used for implementation of a real-world information systems. *Textlligent* system was built as a practical realization of proposed ideas. This section describes the architecture and tests of this system.

Functional requirements of *Texlligent* system can be divided into four groups. User can use the system for preprocessing of set of text documents and to perform the analysis process of sentiments contained in documents with the use of two methods; basing on the classification or using the lexicons. The fourth system functionality is the possibility of hierarchical clustering of documents. Each function is designed with the use of multi-agent architecture, proposed in the algorithms presented in the previous section and the user is separated from the details of system realization.

Figure 6 presents the architecture schema of Textlligent system. The communication between user and system is done by *Command Line Interface*. User can provide necessary parameters and get results with the use of it.

First processing module is *Documents Reading Module* which is responsible for loading the documents. The supervisor of this module is *Document Reading Manager Agent* who synchronizes work of *Document Reading Agents*. *Preprocessing Module* is used by other modules to get the documents after stemization and stop words removal. The effect of this processing module is an input for sentiment analysis modules.

Sentiment Analysis Based on Feature Vectors Classification Module is supervised by *Classification Manager Agent* who delegate *Dictionary Builders Agents*. These Agents are additional units which gather all words needed for building the feature vectors. Next, *Extractors Agents* builds the feature vectors and *Classifiers Agents* perform k-cross validation process. The final result of the module is presented to the user.

Sentiment Analysis Based on Lexicon Module works in the similar way. It has

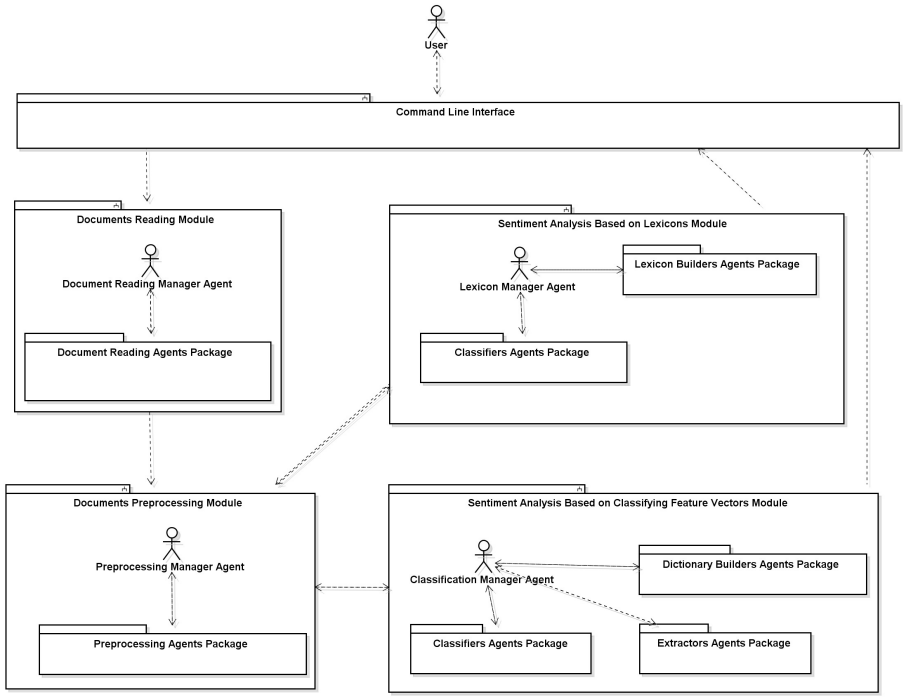

Figure 6: Architecture of *Textlligent* system

its own *Manager Agent* which delegates work to appropriate Agents as it is described in algorithm 2. Manager presents results to the end user of Textlligent system.

The experiments of sentiment polarization analysis for all system's algorithms were performed to test the effectiveness of used methods. To make it possible to compare the results with other experiments the set of movie reviews was used. This set is available in [14]. Set of documents is named *polarity_datasetv2.0* and contains 2 000 labelled text documents. These documents are movie reviews. 1000 reviews are labelled as positive and 1000 are labelled as negative.

The team which prepared this set of reviews tested it with the use of k-cross validation. Set was divided into three parts. The best result was 82.9% of correct classifications. The unigrams were used as the features [14].

Analogous experiments were performed with the use of Textlligent system and its features extractors (TF – term frequency, IDF – Inverse Document Frequency,

Table 1: Results of sentiment analysis of movies reviews done by *Textlligent* system

analysis type	no preprocessing	stemization	stop words	stemization and stop words
Binary	85.23%	**83.88%**	83.38%	**83.33%**
Binary IDF	**85.43%**	82.98%	**84.53%**	81.78%
TF	83.38%	82.88%	80.73%	80.33%
TF_IDF	82.08%	82.88%	81.48%	81.33%
TF_LOG	84.98%	82.28%	83.43%	83.03
TF-IDF_LOG	84.19%	82.38%	83.53%	82.98%
Lexicon	66.39%	61.46%	68.96%	64.52%

Binary – if the word occurs in document its value is 1, otherwise it is 0) and lexicon. The results are presented in table 1. The column *analysis type* stands for feature extractor that was used for the polarization analysis or the lexicon in the second algorithm. The last column *stemization and stop words* contains the results for documents for which the steminization and stop words removal were performed.

The best results were returned for binary extractors with IDF rate. It was 85.43% of correct classifications which means that it is more than 2% better than the best result achieved in [14].

It is interesting that in general the best results where obtained without preprocessing. It means that in sentiment analysis task every information contained in the text can be useful and stemization or stop words removal can cause the quality decrease. What is more, the simple binary extractor was sufficient for obtaining very satisfying results which means that in this task more important is the fact of word occurrence or lack of it. Moreover, the number of particular occurrences of a word is not necessary for the proper polarization of document.

Regarding the results obtained by method based on lexicon they were clearly worse then the results obtained by Classifier Agents. These results can be explained by difficulty of polarization analysis task. Such task is in general ambiguous. Let's consider a review text of action movie. It can contain many negative words like "gun" or "death". The last sentence of review can contain positive opinion but it will be overweighted by negative part of the text. It does not mean that this review is positive or negative. It contains positive opinion but the text is characterized by negative phrases.

6. Conclusions

The analysis presented in the paper confirms that the combination of multi-agent systems theory and intelligent information retrieval and processing methods can be practically used and has many advantages. The research that was carried out proved that combination of multi-agent systems that solves intelligent information retrieval and processing problems improves a designing process of systems of big complexity as well as effectiveness and efficiency of such systems.

Regarding multi-agent approach for designing of information retrieval and processing system, its main advantage is the simplification of building model of such systems. The units-agents are useful for clear presentation and separation of responsibilities of system's elements. This approach is transparent and understandable for human perception. Building very complex systems can became more simple and will minimize the errors during this process. The exchange of communicates between Agents and general communication model of such systems can be easily designed.

On contrary, the implementation of systems designed in multi-agent manner requires more work. It is caused by big number of separated components which must communicate with each other. Additional work on communication protocol is rewarded by simplification of parallelization of processes. The use of Agents helps to design the system properly and gain parallelization out of the box. In Textlligent system Agents work in parallel but implementation details are hidden by their design. Such approach makes the system working much more efficient which is required while processing the huge amount of data. Well designed multi-agent system should predict the parallelization of agents, which makes then independent units.

Despite gain resulting from parallelization, multi-agent approach allows to easily scale the system. It is essential because amount of information needed to be processed is getting bigger every second. Regarding well designed multi-agent systems the scaling should be done without additional implementation costs. The only thing needed to be done is increasing the number of working agents. It allows the users to fully use all of available computational power and return the result as fast as it is possible.

Proposed multi-agent information retrieval algorithms and their practical realizations can be developed to use distributed environment with many huge text document repositories. The agents can work in cloud environments which can be another direction of further research. Universal character of presented approach

makes it possible to adjust it to different requirements. It can be easily done by exchanging methods which are performed by agents. High level of modularity of proposed solutions makes it very simple. Agents can be also designed to automatically test their results and retrained when their effectiveness decreased. Such situation can be extremely important in dynamic environments such as the Internet when documents set is changing rapidly so agents have to adjust themselves.

References

[1] Wooldridge, M., *An Introduction to MultiAgent Systems*, John Wiley & Sons, LTD, Chichester, 2002.

[2] Brenner, W., Zarnekow, R., and Wittig, H., *Intelligent Software Agents. Foundations and Applications*, Springer-Verlag, Berlin Heidelberg New York, 1998.

[3] Weiss, G., *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*, The MIT Press, Cambridge, 1999.

[4] L., P. and Winikoff, M., *An Introduction to MultiAgent Systems*, John Wiley & Sons, LTD, Chichester, 2007.

[5] Chomątek, L. and Poniszewska-Maranda, A., *Modern Approach for Building of Multi-Agent Systems*, LNCS (LNAI), Springer-Verlag Heidelberg, , No. 5722, 2009, pp. 351–360.

[6] Poniszewska-Maranda, A., *Multi-agent system to assure the logical security of data in dis-tributed information system*, Journal of Applied Computer Science, , No. 21(1), 2013, pp. 119–133.

[7] Toru, I., *Mutliagent Platforms. Lecture Notes in Artificial Intelligence 159*, Springer-Verlag, Berlin Heidelberg New York, 1998.

[8] Young, I. C., *Intelligent multiagent application system in an AI system*, Artif Life Robotics, , No. 12, 2008, pp. 6–13.

[9] Azhana, A., Sharifuddin, M. A., Mohd, Y., and , Zaliman, M., *An Exploratory Review of Software Agents*, Information Technology, 2008. ITSim 2008, , No. 3, 2008, pp. 1–8.

[10] Padgham, L. and Winikoff, M., *An Introduction to Multi-Agent Systems*, John Wiley & Sons, LTD, Chichester, 2007.

[11] Manning, C. D., Raghavan, P., and Schuetze, H., *An Introduction to Information Retrieval*, Cambridge University Press, Cambridge, 2009.

[12] Han, J. and Kamber, M., *Data Mining. Concepts and Techniques. Second Edition*, Morgan Kaufmann, San Francisco, 2006.

[13] http://www.coursera.org/course/nlp, *Web course on processing of natural language*, .

[14] Pang, B., Lee, L., and Vaithyanathan, S., *Thumbs up? Sentiment classification using machine learning techniques*, 2002, pp. 79–86.