2016 IEEE 8th International Conference on Intelligent Systems

A Framework for Automated Meta-Analysis: Dendritic Cell Therapy Case Study

Boyko A. A.*, Kaidina A. M.*, Kim Y. C.*, Lupatov A. Yu.*, Panov A. I.†, Suvorov R. E.†, Shvets A. V.†
*Orekhovich Institute of Biomedical Chemistry
of the Russian Academy of Medical Sciences
Moscow, Russia
Email: boyko_anna@mail.ru, alisa.gisina@gmail.com, yankimhcc@gmail.com, alupatov@mail.ru
†Federal Research Center "Computer Science and Control"
of the Russian Academy of Sciences, Moscow, Russia
Email: pan@isa.ru, rsuvorov@isa.ru, shvets@isa.ru

***Abstract*—Increasing amount of scientific publications makes it difficult to conduct a comprehensive review and objectively compare results of previous researches. In some areas of research it is also difficult to extract regularities without computer aid due to complexity of experimental setup and results. Cancer treatment using dendritic cell vaccines is such an area. In this paper we describe a framework for semi-automatic information extraction and further analysis. We also present a case study in the field of dendritic cell vaccination and the corresponding experimental results, which include analysis of separability, classification and regression quality evaluation and cause relations mining.**

***Keywords*—Natural Language Processing, Data Mining, Causal Relations, JSM-method, Genetic Algorithm, AQ-method, Dendritic Cell Therapy, Anticancer Vaccine.**

I. Introduction

Experimental evaluation of scientific ideas is crucial for effective research. Another important part of modern way of research is survey: before doing their own experiments, one has to know what is already done and which methods work best and which of them do not work at all. Also, in many areas of research a scientific paper without experimental evaluation can hardly be treated seriously. Thus, scientific papers are important source of information that has one significant problem: it is difficult to consolidate knowledge about state of the art in some area on the basis of large amount of scientific papers. Needless to say that the number of publications a researcher has to review and compare grows very fast.

In this paper we propose a novel and comprehensive framework to facilitate structured information gathering and comparison. The described framework tries to answer the question "Under which conditions is it reasonable to apply this or that method?".

Also, we present the results of a case study in the field of cancer treatment using dendritic cell vaccines. This field was chosen due to its importance and availability of suitable source data. Dendritic cell (DC) vaccination is an emerging and very promising approach for malignant tumor treatment. Besides, it has some drawbacks: the vaccine has to be prepared specifically for each patient and it is still unclear under which conditions it helps. Dendritic cells are a way to "teach" an organism to kill tumor cells. To be able to do this, dendritic cells have to be prepared using antigens extracted from a tumor. In average, DC vaccination has minor positive impact, but sometimes the effectiveness appears to be much higher. Taking into account these two peculiarities of DC vaccines, it is very important to find out rules that allow to select patients which are more probable to be cured using DCs. One of the main goals of the presented work is to try to build such rules or at least to find a way to do it.

The rest of the paper is organized as follows. In Chapter II we briefly review other researches that aim on creating such rules, in IV we describe how we collected and analyzed the data. Chapter IV also presents experimental results. In Chapter V we sum the research up, analyze pros and cons of the implemented scheme and review possible future work.

II. Related Work

This paper belongs to the intersection of data mining, information extraction and evidence based medicine. Most of the related work review is presented in our previous paper on this topic [1].

Comparative analysis of the DC vaccines applicability was performed earlier in [9], [10]. Criteria used in these papers base on age, sex, disease stage, therapy, comorbidities, biochemical and hematological characteristics of patients, etc. Unfortunately, authors do not provide experimental evidence or reasoning behind some criteria and combinations of patients characteristics.

There are many studies devoted to information extraction from clinical and biomedical texts [11]. Named entity recognition (drugs, diseases etc) is the most popular research direction [12]; relations extraction (e.g. protein interactions) is another well studied area. To make information extraction more precise, various techniques from computational linguistics are used: parsing and semantic analysis, co-reference resolution [11], [13]. The final decision is made on the basis of hand crafted rules or machine-learning classifiers with manually engineered features. An alternative (and more promising) approach is to skip feature engineering step and to learn

978-1-5090-1353-1/16/$31.00

features automatically. For example, deep learning aims on this [14]. Deep learning naturally allows to learn features from graphics (images, video), but text is a graph, not a vector, thus to use deep learning, one has to engineer some initial features anyway. Another way to use deep learning for information extraction is embedding-based [17]. There are also some ready-to-use software systems, such as cTakes [15] and ExaCT [16].

Major drawbacks of the aforementioned methods and systems are inability to extract information from distant parts of texts and difficulties in domain adaptation.

III. Extracting Structured Information

As the primary source of information, we used scientific papers presenting results of experimental clinical research on dendritic cell vaccination effectiveness in application to various malignancies treatment. As a bootstrap corpus and for evaluation purposes, we collected 71 scientific papers.

Then, we extracted information about all patients involved in the research presented in the collected papers. The goal was to get all the information about experiments in vector space representation suitable for further analysis and automatic patient classification. The information extraction framework consisted of the following major steps:

1) Briefly look at the collected papers and describe the structure of data to be extracted.
2) Manually mark up an initial set of patients and extract the rest of information in semi-supervised manner.
3) Map the extracted data to a vector space representation.

The following subsections describe each step in detail.

A. Collected Papers Survey and Data Structure Construction

The goal of this step is to know which characteristics of experiments the collected papers present and how exactly they do this. As a result of this step, the target data structure is defined (speaking object-oriented language, the hierarchy of classes or domain model is composed). We will refer to this data structure as type system. Generally, it must contain all significant characteristics of patients (age, diagnosis, disease stage, tumor markers, previous analyses results, previous treatments, etc.), treatment (how the vaccine is prepared and injected etc.) and the treatment outcome (objective clinical response, analyses after vaccination, survival time, adverse effects, etc.).

We built type system of 25 types with more than 70 attributes. The top level object type is ”Patients group”. Each object of this type describes a patient or a group of them.

The most crucial peculiarity of the data we collected is heterogeneity. Usually, a paper describes involved patients in one or both ways: (a) information about particular patients (explicitly specifying which ones) and (b) information about some groups of patients (not telling which patients exactly are described). We decided to use a single unified type system to handle both of these cases. This is achieved by extending each top-level attribute by two additional ones: Patients Number and Patients Percent (authors of collected papers use one or another of these attributes depending on the situation). Additionally, each top-level attribute except Patients Number may be assigned multiple values (they are of list type). We will discuss the way we use these attributes in section ”Mapping to Vector Space”.

B. Extract Information According to the Type System

To mark the collected papers up and extract information, we used a specifically developed software system previously very briefly described in [1]. This system differs from many others mainly in the following aspects:

- Object-oriented approach to extracted information representation. When the user marks documents up, he or she creates objects according to the previously defined type system and fills in attributes values and connects attributes to the corresponding chunks of text (we will refer to these chunks as cues). The type system allows numeric and string attributes as well as lists of them and objects aggregations and compositions.
- All information stored in the system is indexed in graph database (currently, TitanDB [2]). It allows to treat information extraction as similarity search in the graph and unify algorithms and skip labor-intense feature engineering procedure. There are vertices for each document, object type, object, attribute, normalized value, tokens in the graph. There are also the corresponding edges between them (edges have meaning of ”be a part of” or ”have a value” or ”to correspond to”).

In other words, this software itself *does not use traditional natural language processing techniques*. Instead, it uses such third-party tools as cTakes [15] and Exactus Expert [21] to *populate the graph database and then uses similarity search* to find new relevant text chunks. Target objects in this search are vertices, corresponding to tokens, and the search is carried out on the base of vertices contexts.

This software system has a modern web user interface that allows experts to perform all the workflow. It provides functionality to easily find or import scientific papers, customize type system, label pieces of information and export or analyze the data.

The collected papers contain relevant information about the same patients spread over plain text and one or more tables.

The information extraction process consists of manual initial data set labeling and semi-automatic extraction of the rest data. The goal of the manual labeling step is to provide the bootstrap corpus for training machine learning based classifier to facilitate labeling of the rest papers.

After the initial dataset was completed, the rest papers were analyzed in semi-automatic manner: first, the system found new relevant text chunks (cues) and linked them to attributes of manually created objects; second, for each attribute that had some assigned text we extracted normalized value on the basis of the assigned text, and so on. To find new relevant text chunks, we used an iterative graph-based similarity search

algorithm that relies on adaptive random walks to find vertices that correspond to tokens that are likely to describe particular attributes of patients. After each search iteration, expert assess some of the found chunks and assigns them to attributes of objects. According to the expert's answers, the random walking policy is updated.

The collected papers represent much relevant information in tables, thus it was necessary to recognize and parse tables using a modified algorithm presented in [3].

Types of objects needed for preprocessing steps were merged together into main expert-defined type system and stored in the graph database in the unified way.

Since the used software system is still in active development, we did not evaluate any quality metrics (e.g. precision and recall) for the used algorithms. Also, some important tasks such as automatic object recognition and object linking are not implemented at the moment. However, it could be achieved through extraction of attributes that relate to objects as one-to-one (e.g. "Patient Identifier" attribute).

After all the relevant text chunks were found and assigned to attributes, normalized values were extracted. Normalized value of an attribute is a number or a string or a list of them (depending on the attribute type) that represents the information that the assigned cues state. To normalize values, we used a combination of n-gram based k-nearest neighbors classifier and a set of manually crafted regular expression based rules. Average F1-metric of n-gram KNN varied from 0.6 to 1.0 depending on the attribute (according to 3-fold cross validation over the manually labeled initial data set). To handle cases that KNN misclassified, we created a relatively small number of rules. We also experimented with other machine learning methods (SVM and Random Forest), but on the available data KNN performed best.

As a result of this step, we extracted 927 objects of type "Patients Group" that correspond to 1549 patients total.

C. Mapping to Vector Space

The most crucial peculiarity of the extracted data is its heterogeneity:

- A paper may describe features of particular patients (explicitly specifying which patients which attributes have) as well as abstract groups of patients (not specifying which patients are enclosed in the group). Some features may be mentioned multiple times (in description of patient and descriptions of groups).
- A paper may describe group of patients that intersect or do not intersect.
- Patient groups may be nested into each other.

To sum up, the extracted objects are not immediately comparable. It renders the extracted data inappropriate for application of modern machine learning methods to them "as is".

The next step (Mapping to Vector Space) aims on converting the extracted hierarchical objects to a set of points in a vector space, each of them corresponds to only one patient. To achieve this, we propose to treat each of the extracted "Patients Group" objects as a definition of joint probability distribution of patients features. Thus, to get comparable objects, we can sample from this distribution using an algorithm consisting of the following major steps:

1) Choose a paper that we have not considered yet.
2) Determine the number of patients that this paper describes (as maximal normalized value of "Patients Number" attribute of all "Patients Group" objects extracted from this paper).
3) Create stub vectors, one for each patient.
4) Retrieve all "Patients Group" objects that describe single patients (normalized value of "Patients Number" attribute equals to one), create generators for each of them and uniformly apply generators to the stub vectors.
5) Sort other "Patients Group" objects in ascending order of "Patients Number" attribute value, create generators for each of them and apply.

Generator is a program object which has assigned a probability of being applied to a vector. Generators fill elements in the stub vectors and may be nested into each other. Generators are built using depth-first walk over the objects graph starting from "Patients Group" objects. There are also atomic generators which fill elements in stub vectors corresponding to simple properties of the source objects (numeric, string). Atomic generators cannot have nested sub-generators. Generator is applied to a stub vector using a simple algorithm consisting of the following steps:

1) Check conflicts. If the generator or any of its sub-generators will try to assign a different value to already filled element of the current stub vector, then the generator is considered to be conflicting with the current stub vector and is not applied.
2) If there are no conflicts, "throw a coin" to decide, whether to apply the generator or not. Coins return "apply" with probability N/N_p where N is "Patients Number" value of the current object and N_p is "Patients Number" of the parent object.
3) If the generator is atomic, fill the stub vector element.
4) If the generator is not atomic, invoke all the nested sub-generators.

After we applied the described procedure to the extracted objects, we got a matrix 1549x46. 21 columns represent numeric attributes and 25 represent nominal (or string) ones.

IV. DATA ANALYSIS

During data analysis, we aimed on creating rules that can help to choose patients which with high probability can be successfully treated using dendritic cells vaccination. We also tried to identify possible causes of various treatment outcomes.

To achieve the stated goals, we have conducted a series of experiments with machine learning classification and regression methods to predict: (a) exact objective clinical outcome; (b) group of objective clinical outcome (positive/negative); (c)

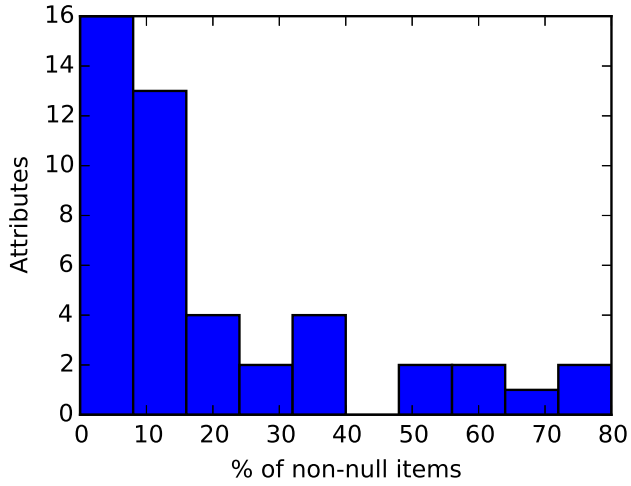
exact survival time; (d) minimal expected survival time (to answer the question "Will patient live more than X days?").

During experiments, we tried two approaches: (a) traditional, using modern machine learning methods (random forest classification and regression); (b) experimental, relying on global optimization for feature selection and logical methods for causal hypotheses construction. We used RandomForestClassifier [20] from Scikit Learn package [4].

A. Data Preprocessing

The input data matrix is filled non-uniformly. The most frequently assigned features: age (80%), Objective clinical response (79%), sex (69%), previous treatment (64%), number of injections (62%). The most rarely assigned features: race ($< 1\%$), numeric values of various markers before and after immunization (about 1%). The overall distribution of fill ratio is present on Figure 1.

Fig. 1. Distribution of filled values percent over attributes



Due to the sparsity and missing values, we preprocessed the input matrix as follows:

1) Remove columns that contain less than $T\%$ filled values. Multiple experiments for various T were conducted.
2) Nominal columns (with values from a small discrete set like $Alive, Dead$) were converted to multiple numeric columns with values from $0, 1$. This procedure is usually referred to as binarization.
3) All the columns were normalized so that all the values belong to $[0, 1]$ range. It was done by subtracting minimal value and dividing by maximal after subtraction. Columns with zero maximal value after subtraction were removed (they are not informative).

B. Predicting Objective Clinical Response

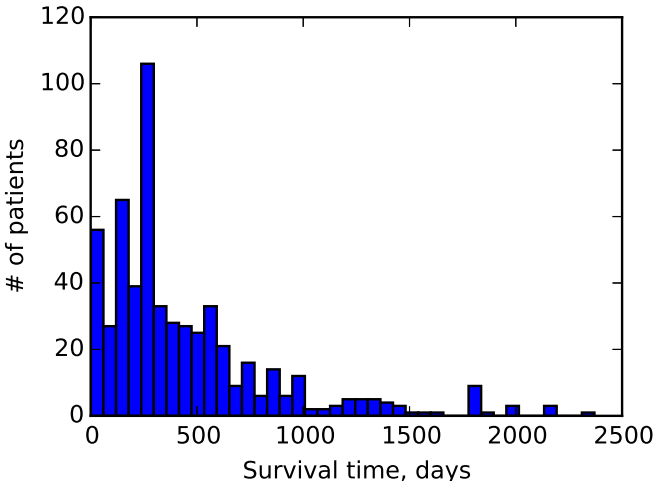
First, we tried to determine the possibility to separate patients with different Objective clinical responses. To answer this question, we used the same data for training and testing. The input matrix contained only 1095 rows with Objective clinical response filled. Other rows were excluded from this experiment. Then, we evaluated the predictive power of the available features. This was done through 3-fold cross validation. The results are presented in Table I.

According to the classifier built, the most important for objective clinical response prediction features were (in importance descending order): number of injections, age, disease stage "1", previous treatment "chemotherapy" and "surgery", number of injected cells, disease "Lung cancer".

C. Predicting Survival Time

Another possible treatment successfulness criterion is survival time (how long a patient lives after treatment). The input matrix contained only 574 rows with Survival time filled. Other rows were excluded from the experiments. Figure 2 presents a histogram of survival time over patients.

Fig. 2. Distribution of number of patients with various survival time



As with objective response prediction, we first evaluate the possibility to predict (test on training dataset), then we evaluate the predictive power of features (3-fold cross validation). The results are presented in Table II.

As we can see, cross-validated results are significantly lower than ideal (first row). This may be due to a number of possible causes: dependency on many other factors that were not present in the available data, power-law distribution of survival time, etc. Figure 3 presents the empirical distribution of number of patients with a particular survival time and objective clinical response (there are 462 patients that have filled both objective clinical response and survival time).

Another way to predict survival time is to guess, if a patient will survive the specified amount of time or not (discriminative approach). We conducted a series of experiments with various survival thresholds from minimal to maximal with step of 100 days. As always, we did separate runs for separability (test on training set) and predictive power (3-fold cross validation) tests. The results are presented on Figures 4, 5 and 6.

TABLE I
RESULTS OF OBJECTIVE CLINICAL RESPONSE PREDICTION EXPERIMENTS (SEP - SEPRABILITY TEST, 3CV - 3-FOLD CROSS VALIDATION)

Objective Clinical Response	# of Positive Examples	F1		Precision		Recall	
		Sep	3CV	Sep	3CV	Sep	3CV
Stable Disease	314	0.63	0.60	0.96	0.88	0.47	0.46
Partial Response	201	0.71	0.69	0.77	0.74	0.66	0.65
Complete Response	108	0.89	0.88	0.98	0.99	0.81	0.79
Progressive Disease	470	0.97	0.82	0.96	0.79	0.98	0.85
Mean (exact response prediction)		0.8	0.75	0.92	0.85	0.73	0.69
Positive Response (Partial or Complete Response)	309	0.8	0.71	0.85	0.88	0.75	0.61

TABLE II
RESULTS OF EXACT SURVIVAL TIME PREDICTION EXPERIMENTS (REGRESSION)

	Explained Variance	Mean Absolute Error, Days	R2 coefficient
Test on training dataset	0.8	96.7	0.8
3-fold cross validation	-0.12	348.6	-0.13

Fig. 3. Distribution of number of patients with various survival time and objective clinical response

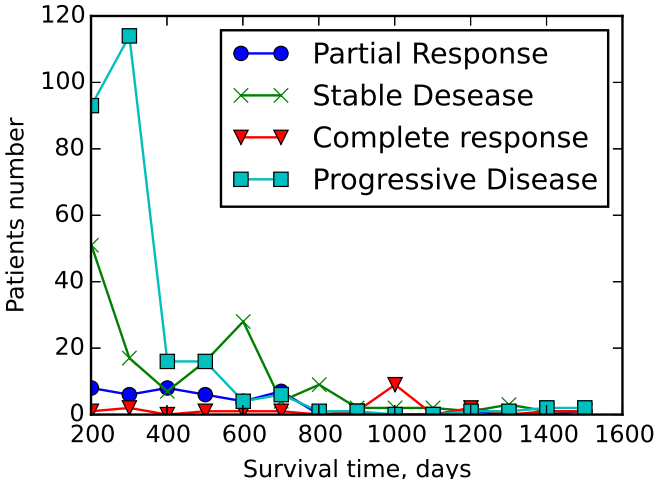
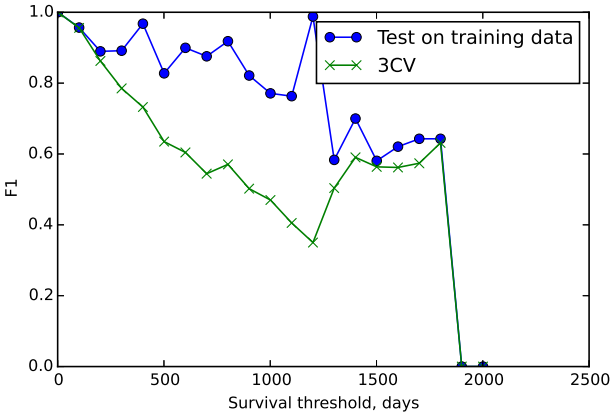


Fig. 4. F1-measure of prediction of survival over each threshold (tested on training data and according to 3-fold cross validation)



According to the classifier built, the most important for prediction of survival excess over threshold features were (in importance descending order): age, number of injections, disease stage "1", total number of injected cells, sex "male", diagnosis "glioblastoma", disease stage "3", DTH antigen "Tumor antigen", sex "female", haplotype "HLA-A*2402".

D. Feature Selection and Cause Relations Mining

As we deal with health care, it is very important to be able not only to predict, but also to explain the predicted value. Explanation usually goes hand-in-hand with cause relations mining. A conventional method for this task is JSM-method [19]. JSM-method is able to pose complex yet stable hypotheses regarding the structure of causes. One significant drawback of JSM-method is its computational complexity. Thus, to be able to apply JSM, we have to reliably select features in a way to not to loose important feature combinations.

To select features, we used a combination of logical inductive learning method AQ (quasi-minimal algorithm) with asymptotic coevolutionary genetic algorithm GAAQ [1]. AQ method builds rules that cover some positive examples and try to not to cover any of negative ones for each class (but not strictly). AQ-rule is a conjunction of disjuncions, each disjunction constraints allowed values of a feature. GAAQ aims on optimizing the feature set by maximizing the number of objects covered by each AQ-rule while minimizing the number of features included in the rule.

Both original AQ and the used GAAQ modification support missing items in datasets. Each missing item is treated as if it would be any of the possible feature values. A rule covers an object if all non-missing features have values included into the corresponding disjunctions. Missing items do not affect

Fig. 5. Precision of prediction of survival over each threshold (tested on training data and according to 3-fold cross validation)

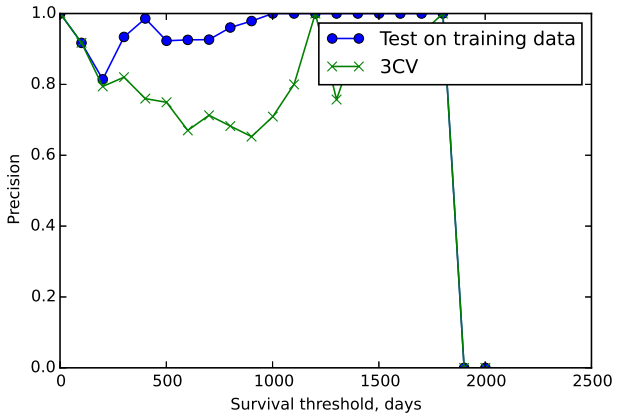
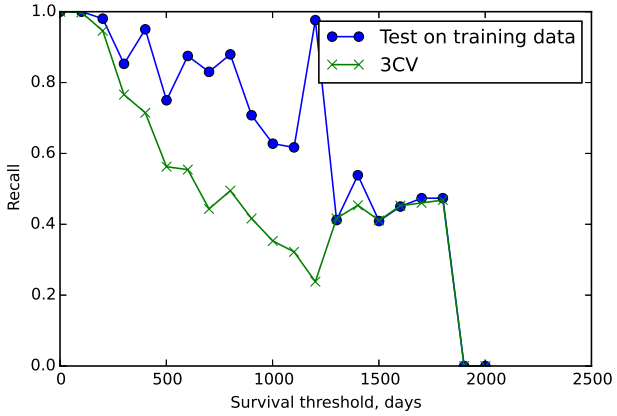


Fig. 6. Recall of prediction of survival over each threshold (tested on training data and according to 3-fold cross validation)



the result. It is obligatory that there is at least one non-missing feature in the object that is constrained in the rule being matched. Table III presents the results of both algorithms applied to the entire dataset.

According to the Table III, GAAQ performs better on the first class than AQ: almost all objects are covered (98%); the rules cover larger number of objects in average (21% vs. 9%); the best found rule also covers larger number of objects (43% vs. 23%). Objects of the second class are better described by the rules found by the AQ, but the average number of covered objects differs slightly.

Characteristic property of rules obtained by AQ is that they are based on features included in maximal coverage rule, which varies from one to another run of the algorithm. While the rules generated by GAAQ do not depend on each other and thus can detect more features that characterize objects of a given class. Also, GAAQ usually generates more complex rules. Therefore these rules and the corresponding features are more suitable for causal relation hypothesis mining.

Feature sets generated by AQ and GAAQ were used to build the input fact base for JSM method [5]–[7]. It is correct to use JSM on the filtered feature sets because the used feature selection algorithms base on the same principle as JSM do (search of maximal objects intersection). The algorithm for fact base construction was presented in detail in [19]. The obtained fact base was preprocessed by conflict elimination and deduplication; when the description of a positive object coincided with a negative one, the last one was removed. Norris algorithm was used to find a maximal intersection [8]. The resulting set of hypotheses was reduced by removing hypotheses which were nested or were too long. Number of found causal relations in various runs is presented in Table IV. From this table one can see that feature selection using GAAQ significantly increases number of causal relations, thus yielding more useful results. One example of cause hypothesis posed by JSM+GAAQ is: a combination of attributes ”Antigen-specific lymphocytes in vitro” - Yes and ”Total number of injected cells” - low and ”Cell maturation inductors” - all except Flt3L and GM-CSF leads to ”Objective Clinical Outcome” - Stable or Progressive. Hypotheses like this have to be carefully reviewed by experts and then clinically verified.

V. CONCLUSION

In this paper we described a novel end-to-end approach for objective comparison of experimental research presented in scientific papers. The proposed framework includes steps for initial information gathering, complex semi-automatic information extraction, postprocessing and final analysis. We also partially developed an integrated software system that implements the framework.

From the applied point of view, we collected and semi-automatically labeled the first dataset (gold-standard corpus) that contains information about more than 70 different experimental researches in the field of cancer treatment using dendritic cell vaccines. Also, we checked a number of hypotheses regarding possibility to predict outcome of treatment on the basis of patient characteristics and description of treatment. The experiments showed that it is possible to predict outcome with relatively low probability of false negative error (high precision) and moderate probability of false positive error (moderate recall). It means that if the classifier says ”yes”, then it is most probably ”yes” indeed, but if it says ”no”, there is significant probability that it is ”yes” in fact. This type of classifier fits the original goal of the project - to provide a rule that allows to select patients that will be successfully treated with high probability. The proposed combination of genetic algorithm-based feature selection and JSM method also performed well.

Despite the results are promising, in order to be used in real life applications, they have to be verified on larger and more complete datasets and checked by experts from the field of dendritic cell vaccination. Possible other directions of further work surely include development of the rest of the software that automates the described framework, its evaluation from

TABLE III
PERFORMANCE INDICATORS OF AQ AND GAAQ ALGORITHMS

Metric	Complete or Partial Response		Death, Stable or Progressive Response	
	AQ	GAAQ	AQ	GAAQ
# of rules	26	75	17	11
Coverage, %	82	98	100	39
% of objects covered by maximal coverage rule	23	43	44	27
Average % of objects covered by one rule	9	21	9	8
Average number of features in one rule	11	14	6	16

TABLE IV
NUMBER OF FOUND CAUSAL RELATIONS

Run	Survival result		Objective response	
	Dead	Alive	Negative	Positive
JSM+AQ	27	12	20	37
JSM+GAAQ (mean)	82	55	81	42

information extraction point of view and application of the whole framework to other domains.

ACKNOWLEDGMENT

The work was supported by the Russian Foundation for Basic Research (Grants no. 13-07-12127-ofi_m and 15-59-31516 RT-omi).

REFERENCES

[1] Lupatov, Alexey Yu, et al. "Assessment of Dendritic Cell Therapy Effectiveness Based on the Feature Extraction from Scientific Publications." (2015).
[2] TinkAurelius. "Titan: A Distributed Graph Database". URL: http://thinkaurelius.github.io/titan/.
[3] Kieninger, Thomas G. "Table structure recognition based on robust block segmentation." Photonics West'98 Electronic Imaging. International Society for Optics and Photonics, 1998.
[4] Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." The Journal of Machine Learning Research 12 (2011): 2825-2830.
[5] Anshakov, O. M., D. P. Skvortcov, and V. K. Finn. "On logical construction of JSM-method of automated hypotheses generation." Doklady Akademii nauk SSSR 320.6 (1991): 1331-1336.
[6] Finn, V. K. "On the definition of empirical regularities by the JSM method for the automatic generation of hypotheses." Scientific and Technical Information Processing 39.5 (2012): 261.
[7] Volkova, A. Yu. "Analyzing the data of different subject fields using the procedures of the JSM method for automatic hypothesis generation." Automatic Documentation and Mathematical Linguistics 45.3 (2011): 127-139.
[8] Kuznetsov, Sergei O., and Sergei A. Obiedkov. "Comparing performance of algorithms for generating concept lattices." Journal of Experimental & Theoretical Artificial Intelligence 14.2-3 (2002): 189-216.
[9] Figdor, Carl G., et al. "Dendritic cell immunotherapy: mapping the way." Nature medicine 10.5 (2004): 475-480.
[10] Murthy, Vedang, et al. "Clinical considerations in developing dendritic cell vaccine based immunotherapy protocols in cancer." Current molecular medicine 9.6 (2009): 725-731.
[11] Aggarwal, Charu C., and ChengXiang Zhai. Mining text data. Springer Science & Business Media, 2012.
[12] Zhou, Li, et al. "Terminology model discovery using natural language processing and visualization techniques." Journal of biomedical informatics 39.6 (2006): 626-636.
[13] Gaizauskas, Robert, et al. "AMBIT: Acquiring medical and biological information from text." Proceedings of the UK e-Science All Hands Meeting. Vol. 14. Nottingham, UK, 2003.
[14] Collobert, Ronan, et al. "Natural language processing (almost) from scratch." The Journal of Machine Learning Research 12 (2011): 2493-2537.
[15] Savova, Guergana K., et al. "Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications." Journal of the American Medical Informatics Association 17.5 (2010): 507-513.
[16] Kiritchenko, Svetlana, et al. "ExaCT: automatic extraction of clinical trial characteristics from journal publications." BMC medical informatics and decision making 10.1 (2010): 1.
[17] Tang, Buzhou, et al. "Evaluating word representation features in biomedical named entity recognition tasks." BioMed research international 2014 (2014).
[18] Neppalli, Kishore, et al. "MetaSeer. STEM: Towards Automating Meta-Analyses." (2016).
[19] A. I. Panov et al. "A Technique for Retrieving Cause-and-Effect Relationships from Optimized Fact Bases". Scientific and Technical Information Processing 6 (2015): In Press.
[20] Breiman, Leo. "Arcing classifier (with discussion and a rejoinder by the author)." The annals of statistics 26.3 (1998): 801-849.
[21] Osipov, Gennady, et al. "Exactus ExpertSearch and Analytical Engine for Research and Development Support." Novel Applications of Intelligent Systems. Springer International Publishing, 2016. 269-285.