4. Cleuziou G., Poudat C. 2008. Proceedings of the 4th DEFT Workshop (Avignon, France, June 8-13, 2008). DEFT '08. TALN, Avignon, France, p. 57–64.

5. DEFT (DÉfi Fouille de Textes) http://deft.limsi.fr/.

6. European Language Recourses Association. DEFT'08 Evaluation Package http://catalog.elra.info/product_info.php?cPath=42_43&products_id=1165.

7. Plantie M., Roche M. and Dray G. 2008. Proceedings of the 4th DEFT Workshop (Avignon, France, June 8–13, 2008). DEFT '08. TALN, Avignon, France, 65–74.

8. Trinh A.-P., Buffoni D., Gallinari P. 2008. Proceedings of the 4th DEFT Workshop (Avignon, France, June 8–13, 2008). DEFT '08. TALN, Avignon, France, p. 75–86.

9. Potter M. A., De Jong K. A. 2000. Cooperative coevolution: an architecture for evolving coadapted subcomponents. Trans. Evolutionary Computation, 8 (Jan. 2000), p. 1–29.

10. Ishibuchi H., Nakashima T. and Murata T. 1999. Trans. on Systems, Man, and Cybernetics, vol. 29, p. 601–618.

УДК 004.93

A NEW METHOD FOR NATURAL LANGUAGE CALL ROUTING PROBLEM SOLVING

T. O. Gasanova[1], R. B. Sergienko[1], W. Minker[1], E. A. Zhukov[2]

[1]Ulm University
43, Albert-Einstein-Allee, Ulm, 89081, Germany
E-mail: taniagasanova@yandex.ru, romaserg@list.ru, wolfgang.minker@uni-ulm.de
[2]Siberian State Aerospace University named after academician M. F. Reshetnev
31, Krasnoyarsky Rabochy Av., Krasnoyarsk, 660014, Russian Federation
E-mail: zhukov.krsk@gmail.com

Natural Language call routing remains a complex and challenging research area in machine intelligence and language understanding. This paper is in the area of classifying user utterances into different categories. The focus is on design of algorithm that combines supervised and unsupervised learning models in order to improve classification quality. We have shown that the proposed approach is able to outperform existing methods on a large dataset and do not require morphological and stop-word filtering. In this paper we present a new formula for term relevance estimation, which is a modification of fuzzy rules relevance estimation for fuzzy classifier. We propose to split the classification task into two steps: 1) "garbage" class identification; 2) further classification into meaningful classes. The performance of the proposed algorithm is compared to several standard classification algorithms on the database without the "garbage" class and found to outperform them with the accuracy rate of 85,55 %. Combination of our approach with 9-NN algorithm for two-stage classification problem definition provides the accuracy rate of 77,11 % for test sample at whole.

Keywords: call classification, term relevance estimation, natural language processing.

НОВЫЙ МЕТОД РЕШЕНИЯ ЗАДАЧИ МАРШРУТИЗАЦИИ ВЫЗОВОВ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ

Т. О. Гасанова[1], Р. Б. Сергиенко[1], В. Минкер[1], Е. А. Жуков[2]

[1]Ульмский Университет
Германия, 89081, Ульм, Аллея Альберта Эйнштейна, 43.
E-mail: taniagasanova@yandex.ru, romaserg@list.ru, wolfgang.minker@uni-ulm.de
[2]Сибирский государственный аэрокосмический университет имени академика М. Ф. Решетнева
Российская Федерация, 660014, Красноярск, просп. им. газ. «Красноярский рабочий», 31
E-mail: zhukov.krsk@gmail.com

Маршрутизация вызовов, основанная на обработке естественного языка, представляет собой сложную и перспективную область исследований в интеллектуальных машинных методах и интерпретации языка. Рассмотрена категоризации пользовательских заявок. Сделан акцент на комбинировании технологий машинного обучения с учителем и без учителя в целях повышения точности классификации. Показано, что разработанный подход способен превзойти существующие алгоритмы на больших базах данных и не требующих морфологического анализа или фильтра в виде «стоп-слова». В предлагаемом подходе осуществляется декомпозиция задачи классификации, к которой сводится маршрутизация вызовов, на две стадии: обнаружение «мусорного» класса и отнесение объектов к значимым классам. Предлагается новая формула оценки релевантности термов при определении значимых классов, которая является модификацией оценки релевантности нечетких

правил в нечетком классификаторе. Используя эту формулу только для 300 наиболее часто встречающихся слов для каждого класса, мы достигли точности классификации 85,55 % при исключении элементов неинформативного класса. Комбинирование предлагаемого подхода с методом ближайших соседей в двухступенчатой постановке задачи обеспечивает точность классификации 77,11 % на всей тестовой выборке.

Ключевые слова: классификация вызовов, оценка релевантности термов, обработка естественного языка.

Natural language call routing can be treated as an instance of topic categorization of documents (where the collection of labeled documents is used for training and the problem is to classify the remaining set of unlabeled test documents) but it also has some differences. For instance, in document classification there are much more terms in one object than in single utterance from call routing task, where even one-word utterances are common.

A number of works have recently been published on natural language call classification. B. Carpenter, J. Chu-Carroll, C.-H. Lee and H.-K. Kuo [1; 2] proposed approaches using a vector-based information retrieval technique, the algorithms designed by A. L. Gorin, G. Riccardi, and J. H. Wright [3] use a probabilistic model with salient phrases. R. E. Schapire and Y. Singer [4] focused on a boosting-based system for text categorization.

The most similar work has been done by A. Albalate, D. Suendermann, R. Pieraccini, A. Suchindranath, S. Rhinow, J. Liscombe, K. Dayanidhi, and W. Minker [5–9]. They have worked on the data with the same structure: the focus was on the problem of big part of non-labeled data and only few labeled utterances for each class, methods of matching the obtained clusters and the given classes have also been considered; they provided the comparison of several classification methods that are able to perform on the large scale data.

The information retrieval approach for call routing is based on the training of the routing matrix, which is formed by statistics of appearances of words and phrases in a training set (usually after morphological and stop-word filtering). The new caller request is represented as a feature vector and is routed to the most similar destination vector. The most commonly used similarity criterion is the cosine similarity. The performance of systems, based on this approach, often depends on the quality of the destination vectors.

In this paper we propose a new term relevance estimation approach based on fuzzy rules relevance for fuzzy classifier [10] to improve routing accuracy. We have also used a decision rule different from the cosine similarity. We assign relevancies to every destination (class), calculate the sums of relevancies of words from the current utterance and choose the destination with the highest sum.

The database for training and performance evaluation consists of about 300.000 user utterances recorded from caller interactions with commercial automated agents. The utterances were manually transcribed and classified into 20 classes (call reasons), such as *appointments*, *operator*, *bill*, *internet*, *phone* or *video*. Calls that cannot be routed certainly to one reason of the list are classified to class *_TE_NOMATCH*.

A significant part of the database (about 27 %) consists of utterances from the "garbage" class (*_TE_NOMATCH*). Our proposed approach decomposes the routing task into two steps. On the first step we divide the "garbage" class into the set of subclasses by one of the clustering algorithms and on the second step we define the call reason considering the "garbage" subclasses as separate classes. We apply genetic algorithms with the whole numbers alphabet, vector quantization network and hierarchical agglomerative clustering in order to divide "garbage" class into subclasses. The reason to perform such a clustering is due to simplify the detection of the class with non-uniform structure.

Our approach uses the concept of salient phrases: for each call reason (class) only 300 words with the highest term relevancies are chosen. It allows us to eliminate the need for the stop and ignore word filtering. The algorithms are implemented in C++.

As a baseline for results comparison we have tested some popular classifiers from RapidMiner, which we have applied to the whole database and the database with decomposition.

This paper is organized as follows: In Section 2, we describe the problem and how we perform the preprocessing. Section 3 describes in detail the way of the term relevance calculating and the possible rules of choosing the call class. In Section 4 we present the clustering algorithms which we apply to simplify the "garbage" class detection. Section 5 reports on the experimental results. Finally, we provide concluding remarks in Section 6.

Problem description and data preprocessing. The data for testing and evaluation consists of about 300.000 user utterances recorded from caller interactions with commercial automated agents. Utterances from this database are manually labeled by experts and divided into 20 classes (*_TE_NOMATCH, appointments, operator, bill, internet, phone* etc). Class *_TE_NOMATCH* includes utterances that cannot be put into another class or can be put into more than one class. The database is also unbalanced, some classes include much more utterances than others (the largest class *_TE_NOMATCH* includes 6790 utterances and the smallest one consists of only 48 utterances).

The initial database has been preprocessed to be a binary matrix with rows representing utterances and columns representing the words from the vocabulary. An element from this binary matrix, a_{ij}, equals to 1 if in utterance i the word j appears and equals to 0 if it does not appear.

Utterance duplicates were removed. The preprocessed database consisting of 24458 utterances was divided into train (22020 utterances, 90,032 %) and test set (2438 utterances, 9,968 %) such that the percentage of classes remained the same in both sets. The size of the dictionary of the whole database is 3464 words, 3294 words appear in training set, 1124 words appear in test set, 170 words which appear only in test set and do not appear in training set (unknown words), 33 utterances consisted of only

unknown words, and 160 utterances included at least one unknown word.

Term relevance estimation and decision rule. For each term we assign a real number term relevance that depends on the frequency in utterances. Term relevance is calculated using a modified formula of fuzzy rules relevance estimation for fuzzy classifier. Membership function has been replaced by word frequency in the current class. The details of the procedure are:

Let L be the number of classes; n_i is the number of utterances of the ith class; N_{ij} is the number of jth word occurrence in all utterances of the ith class; $T_{ji} = N_{ji}/n_i$ is the relative frequency of jth word occurrence in the ith class.

$R_j = \max_i T_{ji}$, $S_j = \arg(\max_i T_{ji})$ is the number of class which we assign to jth word;

The term relevance, C_j, is given by

$$C_j = \frac{1}{\sum_{i=1}^{L} T_{ji}} \left(R_j - \frac{1}{L-1} \sum_{\substack{i=1 \\ i \neq S_j}}^{L} T_{ji}\right).$$

C_j is higher if the word occurs often in few classes than if it appears in many classes.

The learning phase consists of counting the C values for each term, it means that this algorithm uses the statistical information obtained from train set. We have tested several different decision rules defined in table 1.

The best obtained accuracies is achieved with the decision rule C, where the destination is chosen that has the highest sum of word relevancies from the current utterance. But we propose that only terms with highest value of RC (product of R and C) are contributed to the total sum. We have investigated the dependence of the new TRE approach on the frequent words number. The best accuracy rate was obtained with more than 300 frequent words. By using only limited set of words we eliminated the need of stop and ignore words filtering. This also shows that the method works better if utterance includes terms with high C values. This approach requires informative well-defined classes and enough data for statistical model.

Table 1

Decision Rules

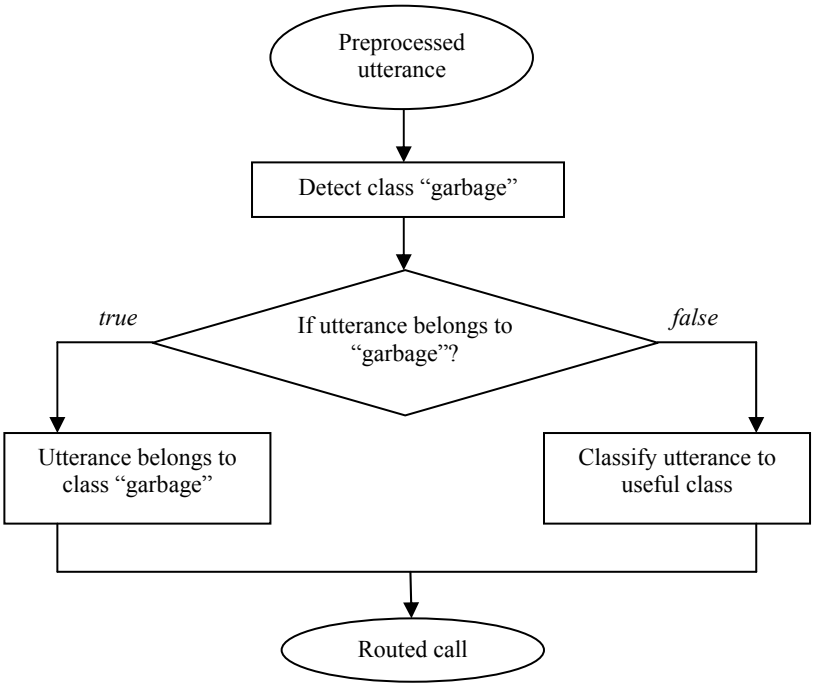
	Decision rules	
RC	$A_i = \sum_{j:S_j=i} R_j C_j$	For each class i we calculate A_i
RC max	$A_i = \sum_{j:S_j=i} \max R_j C_j$	Then we find the number of class which achieves maximum of A_i
C	$A_i = \sum_{j:S_j=i} C_j$	
C with limit	$A_i = \sum_{\substack{j:S_j=i \\ C_j>\text{const}}} C_j$	$winner = \arg(\max_i A_i)$
R	$A_i = \sum_{j:S_j=i} R_j$	

Decomposition of classification problem. "Garbage" class (class TE_NOMATCH) contains 45 % words which do not appear in other classes and portion of "garbage" set is 27 % in the whole database. Furthermore, classification algorithms perform worse on it (it has the biggest classification error). Thus we split the classification task into two problems:

– Detect "garbage" class (class TE_NOMATCH);

– Classify the utterance to meaningful class.

The structure of whole process is given in figure.



Two-stage classification problem

In this case we use special classification method at the first stage. It can be standard classification method or our TRE-approach. For "garbage" class identification we need to unite all informative classes into one class and to teach classification methods for two-class problem definition. For the second stage we need to use only utterances of informative classes from learning sample for classification methods learning.

Clustering methods. After the analysis of the performances of standard classification algorithms on the given database, we can conclude that there exists one specific class (class _TE_NOMATCH) where all standard techniques perform worse. Due to the non-uniform structure of the "garbage" class it is difficult to detect the whole class by the proposed procedure. If we apply this procedure directly we achieve only 55 % of accuracy rate on the test data (61 % on the train data). We suggest to divide the "garbage" class into the set of subclasses using one of the clustering methods and then recount the values of C_j taking into account that there are 19 well defined classes and that the set of the "garbage" subclasses can be consider as separate classes.

In this paper the following clustering methods are used: a genetic algorithm with integers, vector quantization networks trained by a genetic algorithm, hierarchical agglomerative clustering with different metrics.

Genetic Algorithm. The train set accuracy is used as a fitness function. Each individual is the sequence of non-negative integer numbers (each number corresponds to the number of "garbage" subclass). The length of this sequence is the number of utterances from train set which belong to the "garbage" class. We apply this genetic algorithm to find directly the optimal clustering using different numbers of clusters and we can conclude that with increasing the clusters number (in the "garbage" class) we get better classification accuracy on the whole database. We have used the following parameters of GA: population size = 50, number of generation = 50, weak mutation, tournament selection, uniform crossover, averaged by 50 runs. Applying this method we achieve about 7 % improvement of accuracy rate on train data and about 5 % on test data.

Vector Quantization Network. We have also implemented vector quantization network. For a given number of subclasses we search for the set of code vectors (the number of code vectors is equal to the number of subclasses). These code vectors are optimized using genetic algorithm where as a fitness function we use the classification quality on the train set. Each code vector corresponds to a certain "garbage" subclass. The object belongs to the sub-class if the distance between it and the corresponding code vector is smaller than the distances between the object and all other code vectors. Applying this algorithm to the given database we obtain results similar to the results of the genetic algorithm.

Hierarchical Agglomerative Clustering. In this work we consider hierarchical agglomerative binary clustering where we set each utterance to one subclass and then we consequently group classes into pairs until there is only one class containing all utterances or until we achieve a certain number of classes. The performance of hierarchical clustering algorithms depends on the metric (the way to calculate the distance between objects) and the criterion for clusters union. In this work we use Hamming metric and Ward criterion [11].

Table 2

Results of numerical experiments (classification accuracy for test sample)

Method	Learning and classification for all classes, %	Learning and classification only for informative classes, %	Learning for all classes, classification only for informative classes, %
1-NN	74,53	78,85	70,32
2-NN	68,87	75,04	60,26
3-NN	74,16	77,37	68,73
4-NN	72,60	76,75	66,23
5-NN	73,58	76,75	68,39
6-NN	73,05	77,37	67,99
7-NN	73,17	77,54	68,11
8-NN	73,26	77,49	67,99
9-NN	74,04	77,26	68,73
10-NN	73,95	77,43	68,85
15-NN	73,17	76,63	67,71
Bayes with Laplace correction	72,03	76,21	73,62
Bayes without Laplace correction	74,06	76.21	70,84
Bayes (Kernel)	72.03	77,77	73,62
Decision Stump	27,97	73.83	37,92
Rule Induction	40,48	76.21	18,76
Perceptron	21,74	73.83	32,45
Two-stage classification (TRE with clustering + TRE)	76,52	**85,50**	78,17
Two-stage classification (4-NN + TRE)	76,33	**85,50**	**81,18**
Two-stage classification (9-NN + TRE)	**77,11**	**85,50**	78,51

Experimental results. The approach described above has been applied on the preprocessed corpus which has been provided by Speech Cycle company.

We have tested standard classification algorithms (k-nearest neighbors algorithms (k-NN), Bayes classifiers, Decision Stump, Rule Induction, perceptron) and the proposed approach on the database with learning and classification for all classes, with learning and classification only for informative classes, and with learning for all classes and classification only for informative classes. We tested our TRE approach fir two-stage classification problem definition. We use TRE approach with agglomerative hierarchical clustering (it is the best one) for "garbage" class, 4-NN, and 9-NN as classification methods for the first stage or the classification problem.

The results of numerical results you can see in table 2. In this table you can see that our new TRE-approach is the most effective for identification of informative classes. For "garbage" class identification it is better to use k-nearest neighbors algorithms.

Conclusions. This paper reported on call classification experiments on large corpora using a new term relevance estimation approach. We propose to split the classification task into two steps: 1) "garbage" class identification; 2) further classification into meaningful classes. The performance of the proposed algorithm is compared to several standard classification algorithms on the database without the "garbage" class and found to outperform them with the accuracy rate of 85,55 %. Combination of our approach with 9-NN algorithm for two-stage classification problem definition provides the accuracy rate of 77,11 % for test sample at whole.

We can conclude that our approach is appropriate and effective for call routing problem.

References

1. Chu-Carroll J. and Carpenter B. Vector-based natural language call routing, *Comput. Linguist.*, 1999, vol. 25, no. 3, p. 361–388.

2. Lee C.-H., Carpenter B., Chou W., Chu-Carroll J., Reichl W., Saad A., Zhou Q. On natural language call routing, *Speech Commun.*, Aug. 2000, vol. 31, no. 4, p. 309–320.

3. Gorin A. L., Riccardi G., and Wright J. H. How may I help you? *Speech Commun.*, 1997, vol. 23, p. 113–127.

4. Schapire R. E., Singer Y. BoosTexter: a boosting-based system for text categorization, *Mach. Learn.*, 2000, vol. 39, no. 2/3, p. 135–168.

5. Albalate A., Suendermann D., Pieraccini R., Minker W. 2009. *Mathematical Analysis of Evolution, Information, and Complexity*, Wiley, Hoboken, USA.

6. Albalate A., Suendermann D., Minker W. *International Journal on Artificial Intelligence Tools*, 2011, 20(5).

7. Albalate A., Suchindranath A., Suendermann D., Minker W.. 2010. *Proc. of the Interspeech 2010, 11th Annual Conference of the International Speech Communication Association*, Makuhari, Japan.

8. Albalate A., Rhinow S., Suendermann D. 2010. *Proc. of the ICAART 2010, 2nd International Conference on Agents and Artificial Intelligence*, Valencia, Spain.

9. Suendermann A., Liscombe J., Dayanidhi K., Pieraccini R. 2009. *Proc. of the SIGDIAL 2009*, London, UK.

10. Ishibuchi H., Nakashima T., Murata T. Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems, in *Trans. on Systems, Man, and Cybernetics,* 1999, vol. 29, p. 601–618.

11. Ward J. Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association*, 1963, № 58, p. 236–244.

УДК 519.8

DISTRIBUTED SELF-CONFIGURING EVOLUTIONARY ALGORITHMS FOR ARTIFICIAL NEURAL NETWORKS DESIGN

D. I. Khritonenko, E. S. Semenkin

Siberian State Aerospace University named after academician M. F. Reshetnev
31, Krasnoyarsky Rabochy Av., Krasnoyarsk, 660014, Russian Federation
E-mail: hdmitry.91@mail.ru, eugenesemenkin@yandex.ru

In this paper we describe the method of automatic neural network design based on the modified evolutionary algorithms. The main features of the modification proposed are self-configuration and the usage of distributed computing. Implemented algorithms have been tested on the set of classification tasks. The comparison of the genetic algorithm and the genetic programming algorithm's efficiencies is presented.

Keywords: genetic algorithm, genetic programming, self-configuration, distributed computing, artificial neural network, classifiers, automated design.