A Supervised Learning Approach for the Fusion of Multiple Classifier Outputs

C. Ulas, B. Koroglu, C. Bekar, O. Burcak, and O. Agin
R&D and Special Projects Department, Yapi Kredi Bank, Kocaeli, Turkey
Email: {cagdas.ulas, bilge.koroglu, can.bekar, okan.burcak, onur.agin}@yapikredi.com.tr

***Abstract*—In this paper, we propose an alternative method for the fusion of multiple classifier outputs to obtain a potential improvement in the classification performance. Our method mainly operates on decisions of several classifiers, and the individual classifiers, which are trained on different data sets, share common classes and some of them also individually deal with classifying different class labels. For this reason, we first aim to define a common space to represent all the class labels and then propose a supervised learning approach to effectively recognize patterns of the tuples that are formed by concatenating classifiers outputs based on top-*N* rankings. To evaluate the performance of our method, we utilize Random Forest (RF) classifier for this multiclass classification problem and the results demonstrate that our method can achieve promising performance improvement in true positive rate of classification compared to that of the best performing individual classifier yields.**

***Index Terms*—classifier output fusion, random decision forest, supervised learning, decision support systems**

I. Introduction

The main objective of a traditional decision support system is to develop a model that is able to produce correct decisions in a computationally efficient manner with given a small amount of input data [1]. The correctness of decision outputs is very crucial especially in safety critical applications. Due to the limitation of existing individual methods to obtain the best performance in overall accuracy, researchers have come up with a suggestion that one solution for overcoming this limitation might be to combine existing well performing methods, hoping that better results will be achieved. Such fusion of information coming from different sources seems to be worth applying because of the idea that each of individual methods performing on its own data set should produce different errors, assuming that all individual methods perform well, combination of such multiple experts should reduce overall classification error and consequently enable to predict correct outputs.

The problem of classifier fusion has naturally emerged as a need of improvement of classification rates obtained from individual classifiers. Recently, many efforts have been made aiming at combining multiple classifiers into one classification system. Fusion of information obtained from multiple sources can be generally applied on three different levels of abstraction related to the flow of the classification process: data level fusion, feature level fusion, classifier fusion [2]. Among these levels, a vast number of methods or theories have been developed for classifier fusion also referred to as decision fusion. Essentially, the classifier fusion techniques for multiple classifier systems (MCSs) can be divided into two general groups: The fusion strategies generally associated with the first group operate on classifiers and aim to develop a new classifier structure based on those multiple classifiers. These methods [3]-[6] ignore classifier outputs until the combination process finds out a single best classifier or a group of selected classifiers whose outputs are taken as a final decision of entire decision system. The second group of methods [7]-[10] mainly operate on classifier outputs as our proposed approach does, and investigate on new calculation techniques for successful combination of decisions by multiple experts and to produce a single decision.

In this work, we propose a new method for the fusion of multiple classifiers' outputs or decision labels based on a supervised learning approach. Our method treats the classifier outputs simply as the input to a second-level classifier, and in particular exploit Random Forest (RF) algorithm trained on large number of tuples of ranked-outputs obtained from individual classifiers to make prediction on final class decision. Our work significantly differs from previous techniques, for instance voting method [11], simple aggregation operators [12], behavior-knowledge space [13], in several ways: (1) Individual classifiers of the entire system should not necessarily work on totally same class labels, and hence our method allows the fusion of classifiers dealing with different class labels (2) Instead of considering only the most probable decision of classifiers, our approach depends on incorporation of the ranking of best *N* decisions of individual classifiers to obtain a final class decision (3) Though none of the individual classifiers cannot achieve to predict correct class label within top-*N* decision labels, the proposed method has the ability to find correct class label thanks to our approach for reformulating classifier combination to a multiclass classification problem.

The remaining of this paper is structured as follows: In Section II, the proposed multiple classifier fusion approach is presented. We provide a description of the

Manuscript received January 5, 2015; revised May 21, 2015.

doi: 10.18178/ijsps.4.3.198-203

experimental setup with a brief dataset explanation and demonstrate the classification results in Section III, and finally we give concluding remarks in Section IV.

II. Proposed Methodology

The method that we propose in this work purely exploits output classes, which are decided by individual classifiers, and aims to model the relation between tuples and corresponding correct label of an instance in a supervised learning approach. The tuples mentioned here simply denote the representations of outputs of individual classifiers in a ranked way. The top-N decisions of each individual classifier are concatenated and then given as the input to a supervised classifier which can successfully predict the relations between tuples and correct class labels. As shown in Fig. 1, the individual classifiers we have used in this study are trained on different data sets, and therefore possess their own feature vector. In addition to this, the class labels that each individual classifier operates on are not exactly same. Therefore, we first need to represent each class label in a common space and all class labels are represented with associated feature values. In the last step, a supervised learning algorithm is employed on the feature vector representation of classifier outputs to estimate single best class label from whole label space. The flow diagram of our method for the fusion of multiple classifier outputs is illustrated in Fig. 1.

A. Whole Label Space Formation

As it was mentioned before, since some of class labels that individual classifiers are designed to classify are similar and some others vary across individual classifiers, one should form a common label space including all class labels of individual classifiers to enable the development of a multiple classifier system which is able to operate on whole label space and can effectively make prediction on combined classification system. To this end, we propose to reformulate the fusion of multiple classifier outputs as a multiclass classification problem. This will allow us to estimate the correct label of an instance from a larger label set, which may not be possible in the case that all individual classifiers operate on same class labels, as well as will provide the possibility to increase the accuracy of predicting correct classes for the shared class labels among individual classifiers.

Our approach for whole label space formation can be described as follows, also note that same notations are used throughout the paper:

Let our decision system have N_C number of different classifiers operating on their own class label space and let us define a set of S including all individual classifier: $S = \{c_1, c_2, ..., c_{N_C}\}$. Starting from c_1 sequentially, each class label of c_n in S is numbered incrementally from 1 to M, where M is total number of distinct class labels for all classifiers in S. As a result, all common (shared more than one classifier) classes for all $c_n \in S$ are labeled with same numeric value and each distinct class is assigned to an integer in the interval of $[0, M]$. "0" value stands for the empty outputs obtained from any $c_n \in S$, meaning that the individual classifier c_n, if available for this classifier, cannot assign the test instance to any of its class labels.

B. Representation of Classifier Outputs

After forming the label space as stated in Section II-A, output classes of individual classifiers are assigned to an appropriate numeric value from this space. Then, we represent the classifier outputs as a series of elements named as "tuple", each of whose elements include an integer value from whole label space. Suppose that a test instance t_i is classified by each $c_n \in S$, and let us denote the top-*N* class labels obtained from classifier c_i with $(l_{c_{i1}}, l_{c_{i2}}, ..., l_{c_{iN}})$ where each $l_{in} \in [0, M]$, the tuple of an instance t_i is generated by concatenating top-*N* results of individual classifiers and can be represented as follows:

$$T_{t_i} = \left\{ (l_{c_{11}}, l_{c_{12}}, ..., l_{c_{1N}}), ..., (l_{c_{N_C 1}}, l_{c_{N_C 2}}, ..., l_{c_{N_C N}}) \right\} \quad (1)$$

Test instances are given as the input to each individual classifier and the resulted top-*N* outputs of the classifiers are represented with tuples as indicated in (1), aiming at finding the best output from label space. The tuples are represented in $N_C \times N$ - dimensional space.

C. Supervised Learning of Tuples

As opposed to simple techniques in literature, such as majority voting and behavior-knowledge space, to obtain the best final decision from multiple output of classifiers; in this study, we aim to automatically learn the reliability of individual classifiers by calculating the weights of classifier decisions. To do this, we intend to utilize a supervised machine learning algorithm to learn the corresponding weights of features from a training data which includes a large number of tuples and their corresponding true class labels.

The fusion problem here is formulated as a multiclass classification problem, where we target to predict the correct class labels from the tuples of instances. To perform prediction on a large set of tuples, we have used Random Forest (RF) algorithm [14] due to its efficiency in terms of both classification accuracy and computation time for this multiclass problem. Empirically we also attempted to utilize Support Vector Machine (SVM) to compare the performance. However, we have observed that especially applying SVM for multiclass classification with traditional methods, for instance one-versus-one (OvA), or one-versus-rest (OvR), takes too much time during both training to build *M* different models and testing a new instance to predict final class decision. Furthermore, the detection rate of correct classes when using multiclass SVM is slightly smaller than that of RF algorithm achieves.

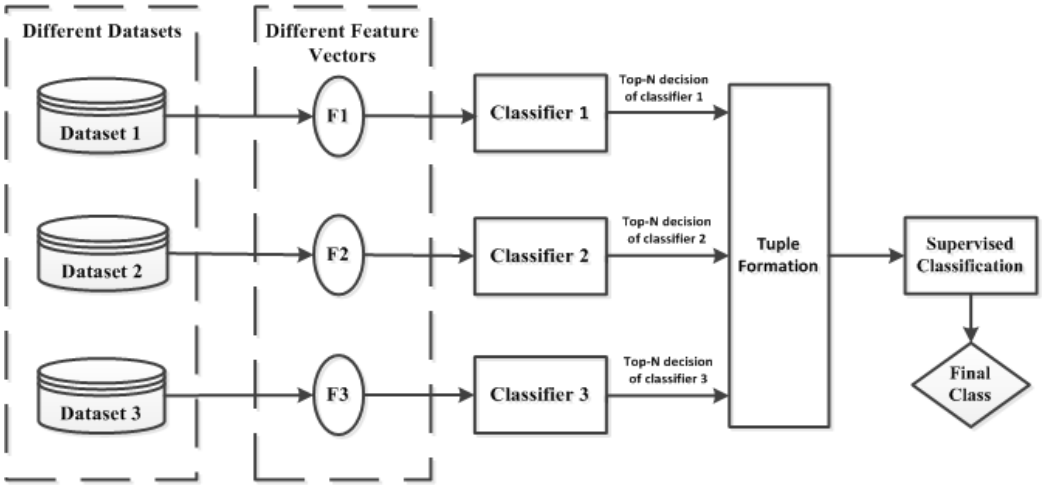

Figure 1. Architecture of the proposed fusion scheme of multiple classifier outputs. Each single classifier is trained on its own data set and top-*N* class labels obtained from these classifiers are directly used for tuple formation procedure. In the end, a supervised classifier operating on multiple classes decides final class.

The number of individual classifiers we employ in this study is almost 3, and the number of classes that each classifier work on classifying is 8, 6 and 12, respectively. The third classifier mainly consists of all the distinct class labels from whole label space, and hence the total number of distinct classifiers operated by all classifiers is 12. Since our second classifier cannot provide a statistical class distribution and is able to estimate only two best class labels, this limits us to work with top-2 decisions of each classifier due to the purpose of maintaining consistency between individual classifiers. Therefore, the dimension of each tuple becomes 6 in our case.

III. Experimental Results

To evaluate the performance of our proposed method, in this study we have used our own data set which is generated by running a batch test on the entire decision support system to obtain top-*N* decisions of individual classifiers. The overall data set is comprised of tuples of decisions with their associated real class labels. After completing batch test, the total number of tuples in data set that we obtained is 8402. As stated before, the dimension of each tuple is 6 in the data and available number of distinct class labels is 12. As shown in Fig. 2, the whole data set is highly imbalanced, where approximately 80% of the data points are covered by only 4 classes. The data set is partitioned into training and test subsets in the ratio of 50%-50%. 10 such partitions are generated randomly for the experiments to effectively generalize the overall performance of the classifier. On each partition, the combination classifier is trained and tested respectively.

For RF classifier, the optimal number of tree is estimated according to the out-of-bag (OOB) classification error metric. The OOB classification errors are calculated using up to 600 decision trees. The number of tree satisfying the lowest OOB error is determined to be used in all training phases of the partitions. For each partition, the maximum number of variable used in each split is empirically fixed to 2 – the smallest integer close to $\sqrt{N_C \times N}$. The performance evaluation of the proposed classifier output fusion method depends on following well-known metrics in literature: Precision, Recall, Accuracy, F1 score. In fact, we use the *weighted average* of these metrics due to the imbalanced class distribution of the data, and final performance of the algorithm on data set is the average of the results over 10 partitions. These metrics are calculated as provided in [15].

Figure 2. Frequency distribution of each class instances in the data set.

TABLE I. Average Performance Values in Percent (%) of Individual Classifiers and Combination Classifier with Various Cases. Metric Values Are Weighted Average of the Values Obtained from Each Class in Classification. Overall Performance Values Are Calculated by Averaging over 10 Partitions.

	Individual Classifiers			Combination Cases				
Metric	Classifier 1	Classifier 2	Classifier 3	Top-1 (All)	Top-2 (All)	Top-2 (Class. 1-2)	Top-2 (Class. 1-3)	Top-2 (Class. 2-3)
Precision	47,33	60,32	**66,74**	73,25	**75,13**	63,28	71,42	71,16
Recall	57,17	**65,78**	64,26	74,90	**76,27**	67,96	73,21	73,85
Accuracy	79,70	83,21	**85,18**	88,71	**89,90**	83,48	87,35	87,76
F1 score	50,54	61,71	**63,21**	73,12	**75,36**	62,45	71,24	71,64

The average performance values for each individual classifier and combination classifier with different cases are shown in Table I. Both the best results for individual classifiers and for various cases of combination algorithm are highlighted with bold fonts in the table. Results in Table I demonstrate that Classifier 3 achieves the best performance among all three individual classifiers in terms of precision, accuracy and F1 score. As mentioned before, this classifier can classify all distinct classes within whole label space, and hence this result is expected. However, Classifier 2 achieves the highest average true positive rate (recall) of classes among all three classifiers since this classifier outperforms other individual classifiers in accurate classification of instances of Class 1 which covers slightly more than half of all instances in test data as depicted in Fig. 1. As compared to other two classifiers, Classifier 1 shows the worst performance and thus can be considered as a weak classifier in this problem.

In this study, we also aim to investigate the performance of our proposed label fusion method by examining various cases including number of individual classifiers used in combination and the number (N) of top-N to comprehend the relation of these variables with classification performance. The right part of Table I shows the average metric values with respect to different cases. As it can be understood by Table I, the best case satisfying highest performance in terms of all metrics is when all three individual classifiers are used with their top-2 classification outputs. The recall rate compared to that of the best individual classifier (Classifier 2) yields is increased by roughly 16% and the corresponding improvement in F1 score from Classifier 3 to this combination case is almost 19.2%. Although the improvement in four metric values is slightly decreased when top-1 (the most probable) decisions are used in fusion algorithm, the increase both in recall and F1 scores is still significant compared to the average performance of the best individual classifier, which is around 10% and 15% respectively for recall and F1 score metrics. This suggests that even reducing the dimension of tuples and taking the single best decisions of individual classifiers into account for fusion procedure may lead to better prediction of the correct class of an instance. Reducing the dimension of tuples also results in a decrease in computation time required for output fusion of multiple classifiers.

Furthermore, the results in Table I indicate that reduction in the number of available individual classifiers results in obtaining worse performance in output fusion. Despite utilizing top-2 classifier decisions of individuals, the performance of combination is greatly affected by the elimination of the best performing individual classifier, which is actually Classifier 3, and only using Classifier 1 and 2 in this case. The results present that with the use of the best performing classifier, the recall, precision and F1 score rates can be improved up to 74%, 71.5% and 71.7%, respectively. However, all of these results are actually lower than the ones obtained by using all individual classifiers, which means that even a weak individual classifier can contribute to overall classification performance by providing additional information to supervised learning of combination algorithm. Our results demonstrate that adding other classifier(s) to the entire system most likely leads to better performance in accurate detection of classes in multiclass classification problem.

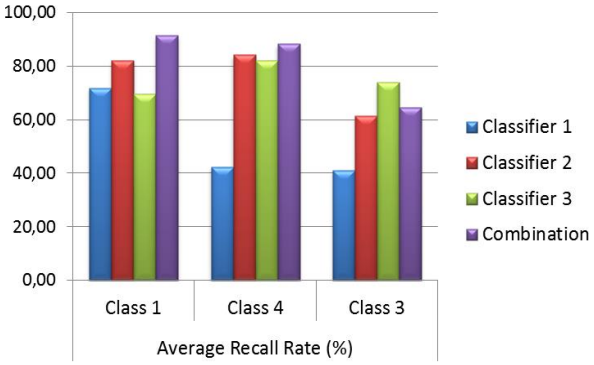

Figure 3. Average recall rates (%) of the most frequent three classes for each classifier in the system.

The average recall rates of the most frequent three classes, which is Class 1, Class 4 and Class 3 respectively as illustrated in Fig. 2, for each individual classifier and combination classifier with the use of top-2 decisions are provided in Fig. 3. The best true positive rates are achieved by combination classifier for Class 1-4 and by Classifier 3 for Class 3. The second best result for Class 3 is obtained by combination classifier with approximately 8% decrease when compared to Classifier 3. The most frequent class in the test set is correctly predicted by combination classifier with a rate of 91.7%, which leads to increase in overall accuracy value as shown in Table I.

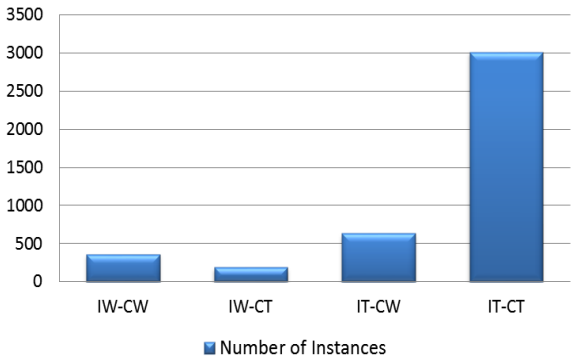

Figure 4. Comparison of the results of individual classifiers and combination classifier, number of instances satisfying following conditions: IW = All of the individual classifiers decide WRONG class, IT = At least one of the individual classifiers predicts TRUE class, CW = Combination classifier decides WRONG class, CT = Combination classifier predicts TRUE class.

To understand both the weaknesses and strengths of our proposed fusion technique in a better way, we applied a strict comparison approach by comparing the number of test instances which are correctly or incorrectly classified by at least one or all individual classifiers and combination classifier. There are totally four different cases with pairs satisfying these conditions for comparison, and the number of test instances satisfying

each case is provided in Fig. 4. These results are obtained by using all three classifiers with their top-2 decisions and averaging over 10 partitions. Approximately 72% of test instances, which are correctly classified by at least one of the individual classifiers within top-2 decisions, are also assigned to true class label by combination classifier. With nearly 6% rate, the proposed fusion technique can achieve to predict correct class label even none of the top-2 decisions of individual classifiers is correct for these instances. This is expected due to the fact that we formulate the fusion of classifier outputs as classification of multiple classes and by employing a supervised classification algorithm at the top level, our approach may enable to predict a different - correct indeed - class which does not exist in the formed tuple. Nevertheless, as a weakness of our method we should conclude from Fig. 4 that the correct classes of roughly 13% of test instances, which are correctly detected by at least one of the classifiers within best two decisions, cannot be predicted by fusion algorithm. In order to evaluate the degree of weakness, we applied a simple majority voting to obtain a single decision within classes presented in tuples for these test instances and we have found out that 84% of them cannot be classified correctly by majority voting, either.

IV. Conclusion and Future Work

In this study, we have described an alternative method for the fusion of classification outputs obtained from multiple individual classifiers. To predict the single best decision of entire system, our method primarily intends to perform the fusion of decisions of several individual classifiers by representing ranked top-*N* classifier outputs with so-called "tuples" and effectively formulating the classifier output fusion as a multiclass classification problem where a supervised classification algorithm can be applied to a large number of tuples with their corresponding true class labels. Our extensive experimental results with various combination cases present that the proposed method achieves to enable potential improvement in the classification performance.

Due to the limitation arising from one of the individual classifier of our decision support system, which does not originally provide more than top-2 decisions, this work cannot present the detailed evaluation of fusion performance with respect to changes in the number (*N*) of best decisions for each individual classifier considered in tuple formation. Therefore, a good future direction of this work can be examining our proposed approach with different individual classifiers for which statistical classifiers may be a good example, enabling further analysis of the effect of top-*N* decisions on the performance of combination classifier in terms of both computation time and overall classification reliability.

Acknowledgment

This work was partially supported by the Scientific and Technological Research Council of Turkey under Grant 3120918 and by YapiKredi Bank under Grant 62609.

References

[1] D. Ruta and B. Gabrys, "An overview of classifier fusion methods," *Computing and Information Systems*, vol. 7, no. 1, pp. 1-10, 2000.

[2] J. C. Bezdek, J. Keller, R. Krisnapuram, and N. R. Pal, *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*, Springer US, 1999.

[3] J. Kittler, M. Hatef, R. P. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226-239, 1988.

[4] K. Woods, W. P. Kegelmeyer, and K. Bowyer, "Combination of multiple classifiers using local accuracy estimates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 4, pp. 405-410, 1997.

[5] R. Benmokhtar and B. Huet, "Performance analysis of multiple classifier fusion for semantic video content indexing and retrieval," *Advances in Multimedia Modeling LNCS*, vol. 4351, pp. 517-526, 2006.

[6] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the EM algorithm," *Neural Computations*, vol. 6, pp. 181-214, 1994.

[7] L. I. Kuncheva, J. C. Bezdek, and R. P. Duin, "Decision templates for multiple classifier fusion: An experimental comparison," *Pattern Recognition*, vol. 34, no. 2, pp. 299-314, 2001.

[8] S. K. Meher, A. Ghosh, B. U. Shankar, and L. Bruzzone, "Neuro-Fuzzy fusion: A new approach to multiple classifier system," in *Proc. 9th International Conference on Information Technology*, Los Alamitos, CA, USA, 2006, pp. 209-212.

[9] J. M. Keller, P. Gader, H. Tatani, J. H. Chiang, and M. Mohamed, "Advances in fuzzy integration for pattern recognition," *Fuzzy Sets and Systems*, vol. 65, no. 2, pp. 273-283, 1994.

[10] L. Xu, A. Krzyzak, and C. Suen, "Methods of combining multiple classifiers and their application to handwriting recognition," *IEEE Trans. Sys. Man. Cyb.*, vol. 22, pp. 418-435, 1992.

[11] L. Lam and C. Suen, "Application of majority voting to pattern recognition: An analysis of its behavior and performance," *IEEE Trans. Sys. Man. Cyb.*, vol. 27, no. 5, pp. 553-568, 1997.

[12] L. I. Kuncheva, "An application of OWA operators to the aggregation of multiple classification decisions," in *The Ordered Weighted Averaging Operators: Theory and Applications*, Springer US, 1997, pp. 330-343.

[13] Y. S. Huang and C. Suen, "A method of combining multiple experts for the recognition of unconstrained handwritten numerals," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 1, pp. 90-93, 1995.

[14] A. Criminisi, J. Shotton, and E. Konukoglu, "Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning," *Foundations and Trends in Computer Vision*, vol. 7, no. 2-3, pp. 81-227, 2011.

[15] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manage.*, vol. 45, no. 4, pp. 427-437, 2009.

Cagdas Ulas was born in Istanbul, Turkey, in 1988. He received the B.S. degree in electronics engineering from Sabancı University, Turkey, in 2011. During his Bachelor's, he also earned a Minor degree in mathematics with high honors. He received the M.S. degree in electronics engineering from Sabanci University serving as a research assistant in Signal Processing and Information System Research Laboratory. Since July 2013, he has been working as a software developer and researcher in R&D department at Yapi Kredi Bank. He has been involved in several on-going projects in his current job.

His research interests lie in the areas of Brain-Computer Interfaces (BCI), Machine Learning, Pattern Recognition, Computer Vision, Information Retrieval and Data Mining. More precisely, he has been fascinated by a broad variety of research areas or fields which mainly practice upon methods in statistical data analysis and machine learning.

Bilge Koroglu (M. Sc.) has received B.S. and M.S. degrees both in Computer Engineering Department, Bilkent University, Turkey. Her research interests include Natural Language Processing, Information Retrieval. She focused on question answering systems which is a subtopic of Natural Language Processing. During graduate works, she conducted an academic research on Turkish search result diversification. For the past two years, she has been working as developer/researcher at Research and Development in Yapı Kredi. She has been working on numerous industrial applications on noisy text classification, short-text clustering and human-computer dialogue systems. Her expertise covers especially on processing human written and unstructured documents.

Can Bekar was born in Turkey in 1988. Bekar has received his B.Sc. in Electrical Engineering from Sabanci University with a Mathematics Minor in 2009 and M.Sc. in Computer Engineering from Koc University in 2013 both in Istanbul, Turkey with a thesis is on concurrent software verification on GPU. He has 3 years of fulltime work experience, 1.5 years as Embedded Software Engineer at a start-up and 1.5 years as Software Engineer at Yapı Kredi Bank R&D in Istanbul. He had interned in IBM Türk during his graduation project, working on a biomedical application with an extension on openCV computer vision library for a grid of PS3's. He has received Microsoft Research and Intel scholarships during his M.Sc. while working as a Research and Teaching Assistant positions at the Research Center for Multicore Software Engineering in collaboration with Microsoft Research and Barcelona Supercomputing Center. (see www.canbekar.com)

Okan Burcak was born in Ankara, Turkey, in 1987. He received the B.Sc. degree in computer engineering from Ege University in 2010 in Izmir, Turkey. He joined the R&D and Special Projects Department of Yapı Kredi in September 2013. Since then he has been working as a senior software developer in R&D department.

Onur Agin received his undergraduate degree from Bilkent University, Ankara, Turkey, in Computer Science Department in 2002. He has worked for 10 years at different engineering positions, now he is the head of R&D and Special Projects Department of Yapı Kredi Bank since 2013.