WISE 2014 Challenge: Multi-label Classification of Print Media Articles to Topics

Grigorios Tsoumakas[1], Apostolos Papadopoulos[1], Weining Qian[2], Stavros Vologiannidis[3], Alexander D'yakonov[4], Antti Puurula[5], Jesse Read[6], Jan Švec[7], and Stanislav Semenov[8]

[1] Aristotle University of Thessaloniki, Thessaloniki 54124, Greece
{greg,papadopo}@csd.auth.gr
[2] East China Normal University, China
wnqian@sei.ecnu.edu.cn
[3] DataScouting, Greece
svol@datascouting.com
[4] Lomonosov Moscow State University, Russia
djakonov@mail.ru
[5] The University of Waikato, New Zealand
asp12@students.waikato.ac.nz
[6] Aalto University, Finland
jesse.read@aalto.fi
[7] University of West Bohemia, Czech Republic
honzas@kky.zcu.cz
[8] Higher School of Economics and the Yandex School of Data Analysis, Russia
stasg7@gmail.com

Abstract. The WISE 2014 challenge was concerned with the task of multi-label classification of articles coming from Greek print media. Raw data comes from the scanning of print media, article segmentation, and optical character segmentation, and therefore is quite noisy. Each article is examined by a human annotator and categorized to one or more of the topics being monitored. Topics range from specific persons, products, and companies that can be easily categorized based on keywords, to more general semantic concepts, such as environment or economy. Building multi-label classifiers for the automated annotation of articles into topics can support the work of human annotators by suggesting a list of all topics by order of relevance, or even automate the annotation process for media and/or categories that are easier to predict. This saves valuable time and allows a media monitoring company to expand the portfolio of media being monitored. This paper summarizes the approaches of the top 4 among the 121 teams that participated in the competition.

1 Introduction

In the past, gathering information was paramount only for top-tier companies. In the information age, mining and categorization of relevant information is necessary for all companies. Media monitoring - the activity of monitoring the

B. Benatallah et al. (Eds.): WISE 2014, Part II, LNCS 8787, pp. 539–546, 2014.

output of the print, online and broadcast media - allows every company to search a wide range of media, from print media to internet publications, and be informed on their area of expertise and remain competitive.

Media monitoring companies rely on human experts that watch/read/listen to media clips and manually index them using concepts from a predefined ontology. This task requires significant amount of time and money to be accomplished. Machine learning can be employed to construct systems for assisting the human annotators by suggesting a list of all concepts by order of relevance, or even for automating the annotation process for media and/or concepts that are easier to predict. This would allow a media monitoring company to invest the saved resources towards expanding the portfolio of media being monitored.

The WISE 2014 challenge was concerned with the task of multi-label classification of articles coming from Greek print media. Data was collected by scanning a number of Greek print media from May 2013 to September 2013. Articles were manually segmented and their text extracted through OCR (optical character recognition) software. The text of the articles is represented using the bag-of-words model and for each token encountered inside the text of all articles, the tf-idf statistic is computed and unit normalization is applied to the tf-idf values of each article. There are therefore 301561 numerical attributes corresponding to the tokens encountered inside the text of the collected articles. Articles were manually annotated by a human expert with one or more out of 203 labels ranging from specific persons, products, and companies that can be easily categorized based on keywords, to more general semantic concepts, such as environment or economy. 99780 articles were collected. The chronologically first 64857 form the training set, and the following 34923 form the test set. The goal is to predict the relevant labels in the test set, where the labels of the articles are withheld. The evaluation metric was the mean F_1 score, also known as example-based F_1 score [1].

This paper discusses the approaches of the top four teams, which were also the ones that provided a summary of the solution. These teams, in order of their final ranking are:

1. Alexander D'yakonov, from the Lomonosov Moscow State University, Russia.
2. Antti Puurula and Jesse Read from University of Waikato, New Zealand and Aalto University, Finland respectivey.
3. Jan Švec, from the University of West Bohemia, Czech Republic.
4. Stanislav Semenov, from the Higher School of Economics and the Yandex School of Data Analysis, Russia.

The rest of this paper is organized as follows. Section 2 discusses the learning approaches employed by the teams to obtain a vector of numerical scores for the labels and Section 3 discusses the thresholding approaches that were used to obtain bipartitions of the set of labels into relevant and irrelevant ones from the numerical scores. Finally, Section 4 presents the conclusions of the WISE 2014 challenge.

2 Learning Methods

The 1st team employed binary relevance instantiated with an ensemble of 10 classifiers:

- Four kNN classifiers, with $k \in \{1, 2, 3, 50\}$ respectively. For $k > 1$ the neighbors are averaged according to their cosine distance. For $k = 50$ initial experiments measured an F_1 score of around 0.68. This was the least accurate family of models.
- Three ridge regression classifiers, with ridge parameter taking values from $\{0.4, 0.8, 1.2\}$. For ridge parameter equal to 0.8, initial experiments measured an F_1 score of around 0.76.
- Three logistic regression classifiers, with L1 regularization and regularization parameter taking values from $\{2, 6, 10\}$. For regularization parameter value equal to 6, initial experiments measured an F_1 score of around 0.78. This was the most accurate family of models.

The ensemble was combined via Stacking [2] using ridge regression. Training data for the meta-model were produced by training the ensemble on the first 50,000 examples of the training set and obtaining its predictions on the rest 14,857 examples. The same last set of examples was used for initial parameter exploration of the base classifiers.

Feature engineering investigations (singular value decomposition, transforming the features to polynomial, adding the features a 2nd time sorted by value) did not lead to significant improvements in the best case.

A distinctive aspect of the approach of the 2nd team was that it reverse-engineered the word counts in the documents from the provided tf-idf vectors. This clever hack allowed the construction of two additional feature vectors: word pairs [3] and 50-300 topics extracted via LDA (Latent Dirichlet Allocation).

The 2nd team created a much larger ensemble, of over 200 classifiers, by employing a variety of multi-label classification algorithms (binary relevance, classifier chains, (pruned) label powerset, random k-labelsets and others) instantiated with a subset of a variety of base classifiers (centroid classifier, multinomial naive bayes, random forest, c4.5, support vector machines) using combinations of the 3 different feature sets. Depending on the size of the feature set and the complexity of the base classifiers different multi-label classifiers could be afforded. In other words not all the above combinations were realized. The final ensemble of the 2nd team consisted of about 50 models selected by a hill-climbing search attempting additions, removals and replacements of the classifiers in the ensemble.

The ensemble of the 2nd team was combined by a variant of Feature-Weighted Linear Stacking [4,5], using the first 59857 documents for developing base classifiers and the following 5000 documents for optimizing the ensemble. This improves simple majority voting by predicting for each instance optimal vote weights based on meta-features computed from all available information. Their variant approximates optimal weights for each training instance, and uses the approximated weights as targets for regression models. Linear regression was used

to develop the ensemble, but for the best submission a random forest with 40 trees was used to predict the vote weights. The set of meta-feature included document features, training set frequencies of the predicted labels and labelsets, as well as correlations between the classifier outputs. One new type of meta-feature that proved useful was the labelset predictions for neighboring documents: since the data was organized in time order, labels occurred often in sequences, and predictions for neighboring documents could be used to improve the classifier vote weight prediction. Two windows of neighboring documents were used: one with 650 documents and one with 6. The score for each label was the sum of the weighted classifier outputs.

The 3rd team employed binary relevance instantiated with a simple linear model trained using stochastic gradient descent (SGD) [6] with modified Huber loss and elastic net regularization. For predicting the posterior probability the method described in [7] was used. The regularization parameters of the binary models were tuned iteratively based on the mean F_1 score: models were tuned one after the other in descending order of frequency. Multiple iterations over all labels were conducted.

In addition, a distinctive aspect of the approach of the 3rd team was that it employed semi-supervised learning. In specific, the aforementioned supervised model tuned using three iterations over all labels was used to give predictions in the test set and then these predictions were taken as ground truth. The same SGD models with another three tuning iterations were applied to this expanded training set. After each tuning iteration, the test data were re-labeled.

The final model arbitrated among: 1) the supervised models, 2) the semi-supervised models, and 3) two additional tuning iterations of the semi-supervised models. This can be seen as a classifier selection approach per label, choosing among 1 supervised model and 3 semi-supervised models, with 3, 4 and 5 tuning iterations respectively.

The 4th team used a linear SVM with L1 regularization for each label. This shows how far one can get by focusing on the design of a powerful thresholding method.

3 Thresholding Methods

Let $\mathcal{Y} = \{\lambda_1, \ldots, \lambda_q\}$ be a set of q labels. Let g_j, $j = 1 \ldots q$ be the predicted score of label λ_j for a given test instance. Let p be a threshold.

The 1st team explored 4 different thresholding rules, according to which a label λ_j was included in the final output when the following corresponding expressions were true:

$$g_j \geq \min(p, \max(g_1, \ldots, g_q)), \tag{1}$$

$$\frac{g_j}{\max(g_1, \ldots, g_q)} \geq p, \tag{2}$$

$$\frac{g_j}{g_1 + \cdots + g_q} \geq p, \tag{3}$$

$$\frac{g_j - (g_1 + \cdots + g_q)/q}{\max_i(g_i - (g_1 + \cdots + g_q)/q)} \geq p. \tag{4}$$

Figure 1 shows the F_1 score of these rules when logistic regression is used as a binary classifier for each label for different values of the threshold p ranging from 0 to 1. The second and forth decision rules appear to be more effective. In its final solution the 1st team used the second rule with $p = 0.55$.

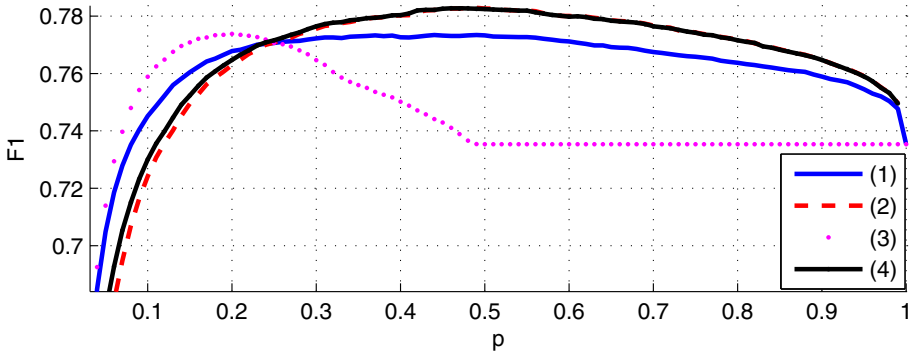

Fig. 1. Performance of decision rules

The 2nd team followed the same simple, but obviously effective, rule, but with $p = 0.5$. This team had successfully used this rule again in the past as part of its winning solution for the LSHTC4 competition [8].

The approach of the 3rd team outputs all labels with probability higher than 0.5. The label with the highest probability is given in the output even if its probability, p^*, is lower than 0.5. In addition, the approach outputs the 4, 3 or 2 labels with the highest probability when the probability of the 4th, 3rd and 2nd label correspondingly are larger than $t_C \cdot p^*$, $t_B \cdot p^*$ and $t_A \cdot p^*$ respectively, where the values of t_A, t_B and t_C are tuned using grid search to optimize the mean F_1 measure on the training data. In cases where multiple conditions are satisfied (e.g. it is possible to assign both two and four labels) the set with the larger cardinality is used.

The approach of the 4th team follows the paradigm of [9,10], which train a regression model to predict a separate threshold for each test instance. This is based on obtaining the predicted probabilities for (a subset of) the training data. Consider for example the predicted probabilities (sorted in descending order) and the corresponding ground truth for an instance of a multi-label learning task with 5 labels that are given in the first two rows of Table 1.

In [9] the regression target was the threshold that minimized the number of labels with wrong prediction, i.e. a threshold in (0.7,0.9) or in (0.3,0.6) in the above example. In [10] the regression target was the threshold that maximized the F_1 score, i.e a threshold in (0.3,0.6) in the above example. Usually, the mean

Table 1. Number of wrong labels (3rd row) and F_1 score (4th row) for 6 different threshold ranges for an instance of a multi-label learning task with 5 labels. The first two rows show the predicted probabilities sorted in descending order and the corresponding ground truth for the 5 labels.

predictions	0.9	0.7	0.6	0.3	0.1	
ground truth	1	0	1	0	0	
wrong labels	2	1	2	1	2	3
F_1	0	0.67	0.5	0.8	0.67	0.57

of the lower and upper boundaries is taken, e.g. for a chosen range of (0.3,0.6), the target would be 0.45. In the approach of the 4th team, two regression models are built, one for predicting the upper and one for the lower boundary of the range that optimizes the F_1 score. A linear combination of the two predicted boundaries can then be taken in the form of $\alpha l + (1 - \alpha)u$, where l and u are the predicted lower and upper boundaries respectively. Figure 2 shows the Mean F_1 for different values of α as investigated by the 4th team. A value around 0.5 leads to best results.

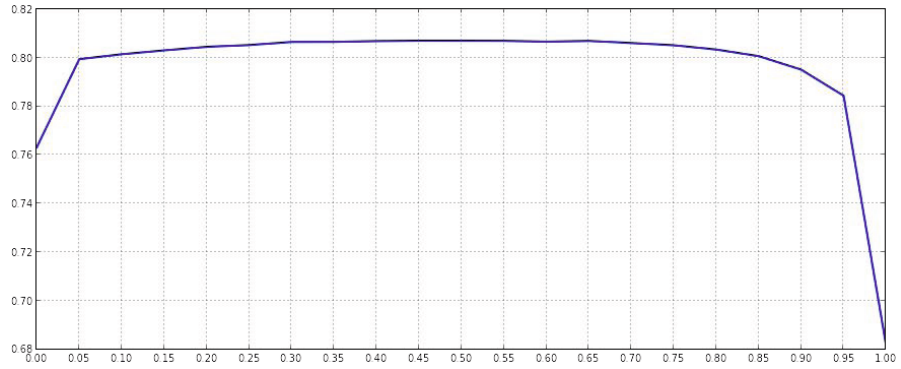

Fig. 2. Mean F_1 for different values of combining the predicted upper and lower boundary

The predicted probabilities were used as input features for the regression in [9], while the features of the instance itself were used in [10]. The 4th team used a versatile set of input features for the two regression tasks: the predicted probabilities, the sorted predicted probabilities, the differences among two consecutive values of the sorted predicted probabilities, the differences among all pairs of the top 10 probabilities and the 20 first principal components extracted by applying PCA to the input space of the main learning task.

4 Conclusions

Examining the top 4 solutions in the WISE 2014 challenge, a number of conclusions can be drawn. A first well-known and expected conclusion is that ensemble methods do well. Indeed, the top 3 solutions employed more than one models (10, 50, 5 respectively). On the other hand, the single model solution of the 4th team shows the importance of a strong thresholding technique. The top 3 solutions used simple, yet effective thresholding techniques, based on dividing each score with the maximum score for a given instance. The 4th team employed a more elaborate thresholding technique that involved learning and appears to be more successful. Another conclusion, more or less also known, is that linear models do well in text classification. Linear models were important components in the solutions of all the top 4 teams. Furthermore, from the approach of the 3rd team, it seems that semi-supervised approaches can go that extra mile compared to supervised approaches, especially for multi-label data where scarcity of labeled examples often arises for some labels.

Another issue worth mentioning is that of overfitting, as it most probably played a decisive role in the final standing. Both the first two teams were well behind in the leaderboard until the private results came out. Both of these teams submitted less than half than the 3rd and 4th teams: (26, 21) vs (52, 72). The first two teams also divided the dataset according to the time-order, instead of doing cross-validation, so that the models were fitted to data closer in time to the test set, and not across the whole dataset. This is always a *wise* choice for data streams.

Comparing the top 2 solutions, we can see evidence in favor of Occam's razor [11]. The solution of the 2nd team involved three different feature representations and over 200 classifiers that resulted from the combination of many different multi-label and single-label classification algorithms, combined by a very powerful stacking variant [4]. The solution of the 1st team involved the original feature vectors and 10 different classifiers from 3 standard families (kNN, logistic regression, ridge regression), combined using standard stacking with ridge regression. Given that they both employed almost the same thresholding strategy, we can say that perhaps simple solutions are still worth being considered first.

Finally, the clever hack of the 2nd team, teaches us that if privacy of the sources has to be protected due to copyright or other issues, then more careful pre-processing has to be applied to the data, such as adding noise, adding bigrams, removing frequent/rare words and disclosing as few details as possible for the actual pre-processing steps.

References

1. Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining multi-label data. In: Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook, 2nd edn., pp. 667–685. Springer, Heidelberg (2010)
2. Wolpert, D.H.: Stacked generalization. Neural Networks 5, 241–259 (1992)

3. Lesk, M.E.: Word-word associations in document retrieval systems. American Documentation 20(1), 27–38 (1969)
4. Sill, J., Takács, G., Mackey, L., Lin, D.: Feature-weighted linear stacking. CoRR abs/0911.0460 (2009)
5. Puurula, A., Bifet, A.: Ensembles of sparse multinomial classifiers for scalable text classification. In: ECML/PKDD - PASCAL Workshop on Large-Scale Hierarchical Classification (2012)
6. Zhang, T.: Solving large scale linear prediction problems using stochastic gradient descent algorithms. In: Proceedings of the Twenty-First International Conference on Machine Learning, ICML 2004, p. 116. ACM, New York (2004)
7. Zadrozny, B., Elkan, C.: Transforming classifier scores into accurate multiclass probability estimates. In: Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2002, pp. 694–699 (2002)
8. Puurula, A., Read, J., Bifet, A.: Kaggle LSHTC4 winning solution. CoRR abs/1405.0546 (2014)
9. Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. In: Advances in Neural Information Processing Systems 14 (2002)
10. Nam, J., Kim, J., Gurevych, I., Fürnkranz, J.: Large-scale multi-label text classification - revisiting neural networks. CoRR abs/1312.5419 (2013)
11. Domingos, P.: The role of occam's razor in knowledge discovery. Data Min. Knowl. Discov. 3(4), 409–425 (1999)