Text Data Management & Processing Assignment Directed Reading Report

LI XUEMENG
NANYANG TECHNOLOGICAL
UNIVERSITY
G2202226E
SINGAPORE
lixu0025@e.ntu.edu.sg

***Abstract*—This is a report on three papers published in the 33rd SIGR 2010: Geneva, Switzerland. The papers are SED: supervised experimental design and its application to text classification [1], Temporally-aware algorithms for document classification [2], and Multilabel classification with meta-level features [3]. The topic for this report is Classification/Automatic Classification.**

Keywords—SED, classification, multi-label, supervised, automatic

I. INTRODUCTION

Classification for text and documents is always a key point and hot issue in the document analysis and processing field. The algorithms used for this task are mostly supervised so that they can predict and allocate unseen documents to certain departments.

Generally, experimental-based active learning methods can derive the distribution of unlabeled datasets and do the one-for-more selection, but Y.Zhen and D.Yeung[1] found that these methods cannot show the importance of some sufficiently labeled data and use them as more as possible. Thus, they introduced a model called Supervised Experimental Design (SED), which would integrate the important labeled data into the experiment at the same time. This has also solved the time-spent problem on the distribution of balanced unlabeled data, which most designs are not good at.

Among the methods inside the Automatic Document Classification (ADC) region, some of them were undisputedly important and famous, for example, KNN, which is one of the most widely-used models for classification. It is a mature method and can be used for both classifying and regression, even for a big-scale or non-linear dataset. Additionally, since it relies on its limited surrounding samples rather than the method of discriminating the category domain to determine the category to which it belongs, so the KNN method is preferred for the sample set to be classified that has more intersections or overlaps in the category domain. However, it still has some corresponding disadvantages. When the sample is unbalanced, the prediction accuracy of rare categories is low. This is also the team [2] interested in. They found that the temporal effects play an important role in ADC, for two or more documents, their weight could not be absolutely equal in most instances, the publisher, time, or authority may influence in different ways. So, they introduced a Temporal Weighting Function(TWF) to help deal with this problem. Also for [3], to solve the pitfalls of low-level feature space learning performance in Multiple-Label Classification (MLC) methods(for KNN is Multi-label KNN), they presented a model which first introduced meta-level features and then apply good learning-to-rank retrieval methods to the space, this model got a good result on learning to rank categories.

Moreover, not only for the example KNN, but the models from the three papers also performed better on the other models like SVM, Naïve Bayes, ML-KNN, and so on. They not only overcome and improved the properties mentioned above, but also did great work in the other fields, but this will also be discussed in *Part 2. Related Work and Development.* For part3, it focuses on the evaluations and some discussion from myself.

II. RELATED WORK & DEVELOPMENT

Text and document classification can be used in many applications/fields, for example, email classifiers, sentiment analysis, Q&A, and so on. Since it is widely used, its data source and the dataset are also diverse. Thus, it is still challenging and meaningful to extract valid information from text.

Active learning is a new branch aiming to find the documents with the most information in them from unlabeled documents and may reduce label-costing. [4] introduced the difference between active learning and traditional learning algorithm, which is active learning, and asks the algorithm to use at least one strategy that this strategy can take control of each iteration, then select one or on set of samples. This model can improve accuracy by marking the most informative sample. Active learning is used when just a few labels, or when it is hard to build a multivariate linear classifier. [5] also found the optimality criterion improves the training by reducing the number of selected examples. Instead of focusing on the selection of data, the team of [1] thinks 'how informatics the labeled text is' is the key point, so as mentioned, they implemented SED on the basis of Transductive Experimental Design(TED)[6]. TED assumes that all unlabeled data is available, so [7] solved the non-convex problem of TED, and the problems of data selection and supervising the model were further updated by SED. The supervised experimental design in SED first took the vector of all absolute values from decision values. Then according to the SVM, SED always wants to find the smallest value among all vectors, which means it has the example with the most information. Thus, they set a control parameter that controls the contribution of label information, once it goes up, the label information would have a larger weight than before. Moreover, the importance of label information has been proved by adjusting the value of the parameter. Also, the importance of candidate size was shown by taking a different percentage from all unlabeled data, and the larger the candidate size, the better the performance is, but more time-costing.

T.Salles and the team[2] had the same idea that different documents have different weights since the factors such as authors or published years which the document has is totally not the same. The solution of the team to this problem was focusing on minimising the influence of temporal. To build the Temporally-Aware Algorithm, the team worked based on some existing models in Automatic Document Classification (ADC): instance selection, instance weighting, and ensemble. These three methods are relative to each other: the method of instance weighting figured out the problem of heuristics may not select desired suitable amount of documents, but it caused the challenge of determining which weighting method and corresponding parameters to use, also the performances would not reach the expectation, Thus, ensembles functions solved this problem by generating various models at the same time. Looking back on the previous works, most of them focused on identifying content [8][9], creating gradual models [10], and improving the different computation problems. [11] introduced a new method using 'window' which focused on the key topic and adjusted the size of the 'window', [12] improved the methods by using three windows so that the collection would be statical and more reliable. Comparing some methods with combined classification models, the team introduced a model which would be easier to manage efficient approaches at the same time. Moreover, the difference between a quite similar ideal [13] and this work was, the stability of the training set produced from [13] could be the key point, it decided which contexts to be classified, and some cases out of the boundary were dismissed since the boundary was decided by sampling from its temporal context in the training data. Hence, scenarios were used to consider both previous and future information, and a temporal weighting function produces a lognormal distribution to avoid only considering the previous information. Based on [13], the new expression connected each term with the stability periods, also a time point has been noted. In the case of repeating representation from one term to multiple time points, they also introduced the temporal distance to all points in one period, with 0 distance for the reference year. Then, the period was represented as a random variable and shows that they are independent of each other.

[3] was focused on multi-label classification(MLC). Two main problems of MLC are: (1) Learn the raking of each input instance's categories. (2) Determine the yes or no choice for each category by setting a threshold on each ranked list. To solve the first problem, [14] used a binary-SVM to learn the scoring from other categories to each independent category. Although binary classifiers were easy to use, they may not have enough global optimization. To solve this drawback, [15] implemented an algorithm called Rank-SVM, this approach maximises the sum of all category margins instead of maximising the margins one by one. Moreover, ML-kNN[16] and IBLR[17] used different probability calculations but the same way to find k nearest neighbors by Euclidean distance. But the final results based on performance on [14] and more methods, IBLR showed a better result than ML-kNN, I suppose the reason could be that IBLR always considered categories under dependencies in pair, and also used logistic regression to show, which ML-kNN just assumed all categories are independent to each other. However, IBLR still used low-level features, this could not be sufficient for MLC. So the team[3] introduced a method to automatically transfer a traditional instance into meta-level features, so that The ordering methods in IR can also be used in MLC, then further improve the optimization problem(the second problem of MLC) mentioned above. To reach the goal, the transformation was the most important hard part. It was defined by the representation of just input instances with a training set of labeled instances. Assuming the meta-level representation as a vector, it should have the most information about the relationship between the instance and the category and would classify the positive and negative instances in the category. Also based on the KNN algorithm, the final meta-level representation could be seen as a combination of L1, L2, cos, and mod vectors, the final vector is a (3k+2)-D vector, with k in the range [10,100]. Each local information represents a feature space, then more feature space could construct discriminative patterns across categories. Moreover, for the learning of ranking categories, the team decided to find a representation of in the pair of instance-category, they chose some existing approaches to concrete examples: RankSVM-IR[18], SVM-MAP[19], LambdaRank[20] and ListNet[21]. Finally, to solve the threshold-learning problem, different threshold strategies were applied to different classifiers. Since binary-SVM and probabilistic binary classifiers all have fixed thresholds, which could not be seen as optimal on the instance-based ranked list. The team used the work from[22], which has a threshold on a per-instance basis, also conditioned on each ranked list. Then the mapping based on this threshold would have an optimal threshold for each list. Also, the choosing-standard for thresholds in [3] was to minimise the False Positive and False Negative. At last, a yes decision was made with scores above or at the threshold, otherwise, the decision would be no, the total of scores was set to one, which improved the performance.

III. Evaluation And Discussion

The team from [1] compared their SED and some other algorithms on Newsgroups Data and Reuters Data to evaluate the performance of the methods. Control parameters and the size of candidates were also set differently. The five methods used to test were SED, Convex SED, Sequential TED, Margin, and Random Sampling. Margin selected data close to the boundary, and Random Sampling means selecting data randomly. From the results, the team showed, for Newsgroup Data with AUC values and curves, SED performed the best and Random Sampling performed the worst, both TED methods performed better than Margin, this showed that using unlabeled data works much better than randomly selected data or just select boundary gathered data. Also, comparing the variance from learning curves of some binary classification tasks showed SED and TED have less chance to choose outliers than Margin. In another evaluation on Reuters Data, SED still performed the best and its learning started earlier than other methods, this indicated that label information is important and affects experimental design positively. Based on the curve results from the paper, In my opinion, when performed on the MCAT Task, once the training sample was over 20, then no matter how many percentages the candidate set taken, their performance did not have an obvious difference than on Autos Task, the reasons may be the training set was not large enough or the distribution was not balanced as expected, thus, when the

original data set has an unbalanced-distribution, SED could not perform well as on balanced distribution. Also, another drawback of SED is when the training set and candidate set are all large, it will take too much time on it and may cause overfitting.

Rocchio, KNN, and Naïve Bayes classifiers were used to cooperate temporal weighting function [2] in two ways: temporal weighting on documents and on scores. For documents, Rocchio and KNN were measured using distance matric but Naïve Bayes was tested as a probabilistic classifier for getting some assumptions. The results showed all three methods can be modified with the function and reproduce algorithms on both documents and scores. Then the original and reproduced three methods were performed on the ACM-DL and Medicine collections. Among all the performances on ACM-DL, the advanced functions all have better results. Especially the improved Rocchio had the best performance in that it raised all two test points including the accuracy of both collections. For KNN, the new one organizes the chosen documents again, so that the more relevant the document is, the closer the position will be(near to the example). On the other hand, the improved Naïve Bayes performed all good except on scores for all collections. This was caused by the lack of information. From my point of view, results shown was just tested on two set of collections, maybe one more or two more collections with larger size could be considered, and this may also increase the accuracy.

[3] was first evaluated on six datasets that have been used in some previous multi-label task experiments, so that the results could be compared to each other. The comparative methods were Binary-SVM [14], Rank-SVM [18], IBLR [17], and ML-kNN [16], which all have been introduced in part2. All methods first calculated the score on a given example and then used the threshold to make a yes/no decision for each instance. For the evaluation of ranked lists and classification decisions, they used metrics. The evaluation results can be concluded below: For the performance between the methods based on 5 datasets and 8 metrics, ListNet performed the best, and IBLR and RankSVM were the least. Focused on the results on RankSVM and ListNet, it showed that meta-level features were useful in learning. Also, some performances showed a rank: MCN-ListNet>IBLR-ML>ML-kNN, the reason for comparing these three methods was they were all instance-based leveraging kNN-based features. However, as mentioned, these test data/methods were also used in previous multi-label tasks. Thus, some of the results from [3] are the same as before. Although the accurate numerical value maybe not be equal, still have the same best/worst performance method-ordering. This proved the reliability of this evaluation, but on the other hand, is it good to have always similar or the same result? And what if testing on some unused dataset, will the result still be solid?
For another evaluation focused on thresholding, the performances of all methods tested have been influenced, the metric Micro-F1 improved but another metric HLoss did not, this showed the thresholding strategy affects the MLC methods. Also from the results, we may know the use of meta-level features and learning-to-rank methods was important to ListNet and can be seen as the main reasons why it performed better than other methods. Based on the results above, the significance test is also done. The decrease in variance proved that all learning-to-rank functions did a better job with meta-level features. As said in Part.2, it also can form a new representation. With this new representation, tests were done to check if the improvement still exists. The variants here were the features, both original and improved meta-level features were evaluated. However, the results showed for ML-kNN and IBLR, the meta-level features may bring badness on performance, and for Binary SVM, poor improvement showed. Thus, the real matter factors were partial-order enhancement and category ranking optimization. Moreover, if the meta-level features have too high dimensions, the machines may be difficult to learn and the advantages of the method will not be shown.

IV. Conclusion

This report has discussed three papers related to the Classification field for text and document processing. They introduced different methods for active learning, automatic document classification, and multi-label classification respectively. The first team introduced a Supervised Experimental Design method to make use of informatic labeled data. The second team implemented a Temporal Weighting Function to consider whether a document is more important or less important. The last team made a transformation from low-level features to meta-level features for multi-label classification problems. All three implementations have made great improvements in their corresponding regions. And also improved reliability in different ways.

V. References

[1] Yi Zhen and Dit-Yan Yeung. 2010. SED: supervised experimental design and its application to text classification. In Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval (SIGIR '10). Association for Computing Machinery, New York, NY, USA, 299–306. https://doi.org/10.1145/1835449.1835501J.

[2] Thiago Salles, Leonardo Rocha, Gisele L. Pappa, Fernando Mourão, Wagner Meira, and Marcos Gonçalves. 2010. Temporally-aware algorithms for document classification. In Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval (SIGIR '10). Association for Computing Machinery, New York, NY, USA, 307–314. https://doi.org/10.1145/1835449.1835502

[3] Siddharth Gopal and Yiming Yang. 2010. Multilabel classification with meta-level features. In Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval (SIGIR '10). Association for Computing Machinery, New York, NY, USA, 315–322. https://doi.org/10.1145/1835449.1835503R.

[4] Cohn, D., Atlas, L. & Ladner, R. Improving generalization with active learning. *Mach Learn* **15**, 201–221 (1994). https://doi.org/10.1007/BF00993277

[5] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. J. Artif. Intell. Res., 4:129–145, 1995.

[6] K. Yu, J. Bi, and V. Tresp. Active learning via transductive experimental design. In ICML, 2006.

[7] K. Yu, S. Zhu, W. Xu, and Y. Gong. Non-greedy active learning for text categorization using convex transductive experimental design. In SIGIR, 2008.

[8] W. W. Cohen and Y. Singer. Context-sensitive learning methods for text categorization. ACM Trans. Inf. Syst., 17(2):141–173, 1999.

[9] N. H. M. Caldwell, P. J. Clarkson, P. A. Rodgers, and A. P. Huxor. Web-based knowledge management for distributed design. IEEE Intelligent Systems, 15(3):40–47, 2000.

[10] Y. S. Kim, S. S. Park, E. Deards, and B. H. Kang. Adaptive web document classification with mcrdr. In ITCC '04, Volume 2, page 476, Washington, DC, USA, 2004. IEEE Computer Society.

[11] R. Klinkenberg and T. Joachims. Detecting concept drift with support vector machines. In P. Langley, editor, ICML '00, pages 487–494, Stanford, US, 2000. Morgan Kaufmann Publishers, San Francisco, US.

[12] M. M. Lazarescu, S. Venkatesh, and H. H. Bui. Using multiple windows to track concept drift. Intell. Data Anal., 8(1):29–59, 2004.

[13] L. Rocha, F. Mourao, A. Pereira, M. A. Gon¸calves, and W. Meira Jr. Exploiting temporal contexts in text classification. In Proc. of the CIKM '08, 2008.

[14] Vapnik, V. (2000). The nature of statistical learning theory. Berlin: Springer.

[15] Elisseeff, A., & Weston, J. (2001). Kernel methods for multi-labelled classification and categorical regression problems. In Advances in neural information processing systems (Vol. 14, pp. 681–687). Cambridge: MIT Press.

[16] Zhang, M., & Zhou, Z. (2007). ML-KNN: A lazy learning approach to multi-label learning. Pattern Recognition, 40(7), 2038–2048.

[17] Cheng, W., & Hüllermeier, E. (2009). Combining instance-based learning and logistic regression for multilabel classification. Machine Learning, 76(2–3), 211–225. doi:10.1007/s10994-009-5127-5, http://www.springerlink.com/content/m20342966250233x/.]

[18] Joachims, T. (2002). Optimizing search engines using clickthrough data. In Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining (pp. 133–142). New York: ACM

[19] Yue, Y., & Finley, T. (2007). A support vector method for optimizing average precision. In Proceedings of SIGIR07 (pp. 271–278). New York: ACM.

[20] Burges, C., Ragno, R., & Le, Q. (2007). Learning to rank with nonsmooth cost functions. Advances in Neural Information Processing Systems, 19, 193.

[21] Cao, Z., Qin, T., Liu, T., Tsai, M., & Li, H. (2007). Learning to rank: from pairwise approach to listwise approach. In Proceedings of the 24th international conference on machine learning (p. 136). New York: ACM

[22] Elisseeff, A., & Weston, J. (2001). Kernel methods for multi-labelled classification and categorical regression problems. In Advances in neural information processing systems (Vol. 14, pp. 681–687). Cambridge: MIT Press