Automatic Multilabel Categorization using Learning to Rank Framework for Complaint Text on Bandung Government

Ahmad Fauzan
School of Electrical Engineering and Informatics
Institut Teknologi Bandung
Bandung, Indonesia
ahmad.fauzan@outlook.com

Masayu Leylia Khodra
School of Electrical Engineering and Informatics
Institut Teknologi Bandung
Bandung, Indonesia
masayu@stei.itb.ac.id

***Abstract*—Learning to rank is a technique in machine learning for ranking problem. This paper aims to investigate this technique to classify the responsible agencies of each complaint text of LAPOR, which is our government complaint management system. Since this categorization problem is multilabel one and the latest work using learning to rank for multilabel classification gave promising result, we work on experiment to compare the typical classification solution with our proposed approaches on this multilabel categorization problem. The experiment results show that LamdaMART, which is listwise approach in learning to rank, is the best algorithm for classifying the primary agency and the secondary agencies for complaint text.**

Keywords — learning to rank; multilabel classification; government; machine learning; complaint management; text categorization

I. INTRODUCTION

Automated text categorization has witnessed a booming interest in the last 10 years, due to the increased availability of documents in digital form and the ensuing need to organize[1], including complaint texts on government agencies. In Indonesia, especially Bandung, complaints can be informed to government agencies in two different ways. First, people post their complaints on social media, such as Twitter, by mentioning the right person or account to inform government. Second, people submit their complaints on internet through complaint management system. In Indonesia, LAPOR (Layanan Aspirasi dan Pengaduan Online Rakyat) [2] is the first online system in Indonesia for receiving complaints [3]. Our main focus on this paper is complaint texts on LAPOR.

LAPOR depends on some system administrators who will manually read each submitted complaint text, deliver it to a government agency that should handle the complaint, and make a copy to some government agencies that have relation to the complaint but not responsible to handle it directly. If the data stream of complain text is huge, this will not only consume time at least three days for each complaint, but also sensitive to inconsistency.

In this paper, we propose automatic text categorization for managing complaint text. Automatic text categorization is an activity to labeling natural language text into predefined categories [1]. If a text is categorized into only one category, it is called single label categorization, but if a text is categorized into one or more categories, it is called multilabel categorization (also known as multilabel classification).

LAPOR has unique structure consisting of two labels. First label is for the agency that is responsible to handle the complaint text, and second label is for some agencies that have connection to the complain text. The typical solution will classify each text into two steps. First, the text will be classified to get the primary agency that is responsible on the problem. Then, the text will be classified again to get the secondary agencies that have connection to the complaint text.

This paper will investigate learning to rank, which is a technique in machine learning that is used to train a model for ranking [4]. Learning to rank can be employed in a wide variety of applications in Information Retrieval (IR), Natural Language Processing (NLP) and Data Mining (DM). Typical applications of learning to rank are document retrieval, expert search, definition search, collaborative filtering, question answering, keyphrase extraction, document summarization and machine translation [5].

The main idea of using learning to rank for this problem is to rank priorities of all agencies based on complaint text. There are two approaches in using learning to rank. First, we apply multilabel classification to select relevant agencies to the complaint text. After that, we rank the classification result to get rank 1 as the primary agency and remaining rank as secondaries agencies. The second approach does not use the multilabel classification, and only employs learning to rank. This approach ranks all agencies, take the first rank as primary agency, and then using threshold in the ranking to get secondary agencies .

In the following sections, we will discussed related works. Then, we will explain our method for this problem in section III. In section IV, we will explain the analysis of used data. In section V, we will explain how we do the experiments. The result obtained is explained in section VI, and we give more consideration in section VII. The last section contains conclusion and further works of the proposed method.

II. RELATED WORKS

There are some techniques that can be used in multilabel classification. One of them is by transforming the problem into single label case. In this technique, the text is classified into binary (yes or no) problem for each category. This includes binary relevance and label powerset. In binary relevance, dataset is transformed into some datasets that has only one label, and each dataset is trained with each predefined label [6]. Different from binary relevance, label powerset is simpler but more effective. Label PowerSet considers the label correlation and labels is transformed into label set and using that label set to train the dataset [7].

Learning to rank is a technique in machine learning used to train a model for ranking [4]. Learning to rank has three approaches for learning, consisting of pointwise, pairwise and listwise. In pointwise approach, the ranking problem is transformed into classification, regression, or ordinal classification of the ranking as rank label including OC SVM [8], MART [9] and Random Forest [10]. In pairwise approach, ranking is transformed into pairwise classification or regression, including RankSVM [11] and RankNet [12]. A classifier classifies the ranking order of ranking pair. In listwise approach, ranking lists are used as instances in both learning and prediction. The group structure of ranking is maintained and ranking evaluation measures can be more directly in corporate into the loss functions in learning [4]. The listwise approach is including ListNet [13], AdaRank [14], and LambdaMART [15].

Evaluation on performance of ranking model is carried out by comparing the ranking lists output and the ranking lists given as ground truth [4]. These include NDCG (Normalized Discounted Cumulative Gain) [16], DCG (Discounted Cumulative Gain) [16], and MAP (Mean Average Precision).

Latest work showed that visibility in using learning to rank as multilabel classification by Yang & Gopal [17]. They conducted learning to rank as framework to rank all categories based on each instance. They employed meta-level feature for each instance. Meta-level feature shows distance between instance and category. In principle, with help instance-category feature, learning to rank algorithm is employed as plugin for ranking in MLC. To enable classification decisions in multilabel classification, this technique applied a threshold to the ranked list of categories for each instance. Their experiment used various dataset such as *Emotions*[18], *Scene*[19], *Yeast*[20], *Citeseer*, *Reuters-21578* and *Vowel*[21] by using meta-level features. This method has outstanding result than other general methods of classification such as IBLR [22], ML-kNN[23], Binary-SVM [24] and RankSVM [20].

III. OUR APPROACHES

This paper proposes two approaches to solve this problem. First approach is combining multilabel classification and learning to rank. Multilabel classification will be used for filter relevant categories, then learning to rank will be used for ranking the result categories of multilabel classification. By using same training set, we train multilabel classifier and learning to rank model. Fig 1. shows how this approach do learning and predicting the agencies.

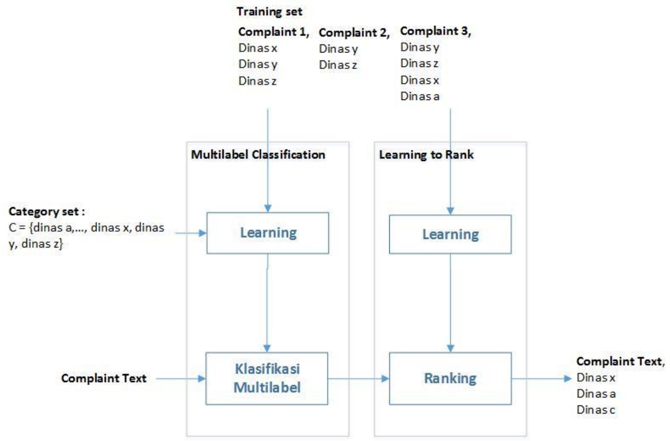

Figure 1. Combination of Multilabel classification and Learning to rank. The word "Dinas" means government agency.

Second approach is using learning to rank for multilabel classification as Yang & Gopal[17] did. In this approach, we do not use meta-level feature, but we apply typical feature for text categorization and add one feature for distinguish between the categories. In learning, score lists will be used to determine the threshold score for making classification decisions. Fig 2. shows how this approach learn and predict.

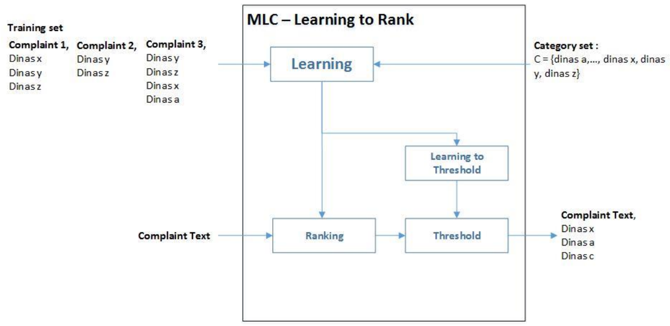

Figure 2. Learning to rank for Multilabel Classification. The word "Dinas" means government agency

IV. DATA ANALYSIS

We got dataset from UKP-PPP, which is government institution that has responsible to manage LAPOR [2]. Our dataset only consists of complaint texts on Kota Bandung Government . This dataset has 2230 instances, and 72 categories of agencies. Each instance consists of 4 fields, i.e. id, complaint text, topic and label.

Since each category has different number of instances, this dataset is imbalanced. For example, agency named "Dinas Bina Marga dan Pengairan (DBMP) Kota Bandung" has 622 complaint texts, and "Dinas Perhubungan (Dishub) Kota Bandung" has 335, but some agencies only have 1 complaint text.

Mostly, complaint text uses Indonesian language. But sometimes, Sundanese words appear in some sentences. Based on this fact, the extraction feature will based on term frequency in training set and text preprocess will based on Indonesian terms such as stopword removing. The preprocess will delete all Sundanese words, such as "mah", "kumaha", "tos", etc.

Basically, our features are all unique terms of complaint text. We apply three types of term weighting, including binary (exist or not), term frequency (tf), and tf-idf. We also employ more features for heuristic. There are two more features considered to be investigated. First is the topic of the complaint text and second is existence of agency name in complaint text. Since the target government agencies are sometimes inside the complaint text, it is our second additional feature that is considered to use in this paper.

V. EXPERIMENT

Our experiment used training set and testing set from given dataset. Testing size consists of 1000 instances, and training size consists of 1230 instances. All categories must be consisted in both training and testing set.

The experiment has three scenarios. First scenario is the base process that uses single-label classification and multilabel classification. Second scenario used combined learning process from multilabel classification and learning to rank. There are two steps in this scenario. First, multilabel classifier will classify complaint text into relevant categories. Then, the result labels will be ranked using learning to rank to get primary agency and secondary agencies. Third scenario used the approach from Yang & Gopal [16]. For threshold, this scenario will use average of threshold scores from training set.

In each scenario, there are three sub-scenario. This sub-scenario will use combination of features will be used, i.e. vector of terms, vector of terms and topic, and vector of terms and existence of label inside complaint text.

The single-label classification used three algorithms including Artificial Neural Network (ANN), Naïve Bayes, and Support Vector Machines (SVM). We applied multilabel classification Binary-J48, Binary-SVM, Binary-Naive Bayes, Label PowerSet (LP)-kNN, LP-NaiveBayes, LP-J48 and ML-kNN. We also used Learning-to-Rank algorithms RankNet, ListNet, Random Forests, MART, and LambdaMART. For evaluation of learning to rank, the measurement used is NDCG@5.

We employed Weka [25] for single label classification, MULAN [26] library for multilabel classification, and learning to rank library in Java named RankLib [27].

VI. RESULTS

We obtain four sets of results from our experiments. The first set focuses on first scenario of experiment which use single label and multilabel classification (TABLE I). The second set focuses on second scenario of experiment which use the combination of multilabel classification and learning to rank (TABLE II). The third focuses on third scenario of experiment which only use only learning to rank (TABLE III). The fourth set focuses on comparison of three scenarios before by taking the best algorithm each scenario (TABLE IV). We discuss the result sets in the following subsections.

A. *Performance of single-label and multilabel classification*

This scenario has two steps of classification. Step 1 is used to get the primary agency and step 2 is used to get the secondary agencies. This two steps do not have independencies between them. It means we can take the single-label algorithm from one sub-scenario and multilabel algorithm from another sub-scenario.

As shown by Table I, the best algorithm for single label classification in this problem is SVM except for term features, Naive Bayes classifier outperforms SVM. The best performance for single label classification is using count terms feature and complaint topic and SVM as algorithm. Otherwise, for multilabel classification, best performance is using ML-kNN, terms feature and existence label in text with binary representation of each term.

TABLE I. THE EXPERIMENT RESULT OF FIRST SCENARIO

Step	Algorithm	Binary	Count	TF-IDF
Vector of terms				
Single-Label	NaiveBayes	**55.46**	42.20	**55.76**
	J48	51.25	50.90	46.85
	SVM	55.26	**55.60**	53.85
Multi-Label	MLkNN	28.87	16.44	32.71
	BR-Naïve Bayes	41.23	37.81	46.66
	BR-J48	40.33	43.44	44.82
	BR - SVM	49.03	50.59	46.93
	LP - kNN	36.66	37.37	40.87
	LP - NaiveBayes	47.13	48.72	**55.23**
	LP - SVM	**55.40**	**57.06**	52.40
	LP-J48	48.49	43.39	43.36
Vector of terms and complaint topic				
Single-Label	NaiveBayes	64.13	52.74	**63.63**
	J48	56.91	55.92	56.05
	SVM	**65.11**	**65.65**	62.83
Multi-Label	MLkNN	**71.75**	**71.70**	**71.80**
	BR-Naïve Bayes	52.72	44.37	54.92
	BR-J48	71.37	71.35	71.35
	BR - SVM	69.42	66.73	65.69
	LP - kNN	64.78	63.65	65.60
	LP - NaiveBayes	57.93	52.00	57.73
	LP - SVM	70.20	67.79	67.21
	LP-J48	71.57	71.55	71.45
Vector of terms and existance label in text				
Single-Label	NaiveBayes	62.87	51.75	61.57
	J48	55.87	55.29	54.48
	SVM	**63.95**	**64.35**	**61.79**
Multi-Label	MLkNN	**72.33**	**72.30**	**71.90**
	BR-Naïve Bayes	57.20	43.82	54.79
	BR-J48	71.00	70.90	70.90
	BR - SVM	68.39	65.78	64.58
	LP - kNN	62.98	61.47	63.82
	LP - NaiveBayes	55.09	51.64	57.28
	LP - SVM	69.14	68.24	64.78
	LP-J48	71.36	71.05	71.10

B. *Performance of multilabel classification and learning to rank*

Like first scenario, this second scenario has two steps. Step 1 is to get the relevant agencies of complain text and step 2 is to rank the previous step to get the primary agency and the secondary agencies.

As shown by Tabel II, the best performer of multilabel classification on this scenario is Label PowerSet–SVM with terms and complaint topic feature with binary representation of term.

All ranking algorithms show a great performance on the relevant categories. In this experiment, we apply NDCG@5.

In fact, this scenario must be included the multilabel classification and learning to rank on same scenario. Based on this fact, the best performance of this scenario is using LP-SVM and Random Forests with binary terms and complaint topic as features.

TABLE II. THE EXPERIMENT RESULT OF SECOND SCENARIO

Step	Algorithm	Binary	Count	TF-IDF
Vector of terms				
Multi-Label	MLkNN	28.87	16.44	32.71
	BR-Naïve Bayes	41.23	37.81	46.66
	BR - J48	42.33	43.44	44.82
	BR - SVM	49.03	50.59	46.93
	LP - kNN	36.66	37.37	40.87
	LP - NaiveBayes	47.13	48.72	**55.23**
	LP - SVM	**55.40**	**57.06**	52.40
	LP - J48	48.49	43.39	43.36
Ranking	MART	**97.35**	**96.71**	**97.73**
	RankNet	96.61	96.04	96.53
	ListNet	96.66	96.05	96.67
	Random Forest	97.78	97.20	97.51
	LamdaMART	97.31	96.61	97.44
	Cordinate Ascent	96.16	95.47	96.17
Vector of terms and complaint topic				
Multi-Label	MLkNN	39.46	40.06	41.46
	BR-Naïve Bayes	49.53	39.54	48.69
	BR - J48	45.02	44.49	44.64
	BR - SVM	53.21	52.89	49.21
	LP - kNN	43.44	43.56	44.33
	LP - NaiveBayes	58.57	50.17	**58.30**
	LP - SVM	**59.12**	**58.63**	55.48
	LP - J48	46.68	46.83	46.80
Ranking	MART	97.30	97.34	97.38
	RankNet	96.51	96.55	96.63
	ListNet	96.66	96.66	96.66
	Random Forest	**97.43**	**97.47**	**97.49**
	LamdaMART	97.19	97.19	97.18
	Cordinate Ascent	96.18	96.18	96.18
Vector of terms and existance label in text				
Multi-Label	MLkNN	34.61	18.41	32.64
	BR-Naïve Bayes	47.03	37.16	46.79
	BR - J48	44.05	42.53	44.02
	BR - SVM	50.80	50.79	47.01
	LP - kNN	38.56	33.76	41.81
	LP - NaiveBayes	**57.66**	49.00	**57.44**
	LP - SVM	56.59	**56.94**	53.45
	LP - J48	42.65	42.58	42.27
Ranking	MART	97.33	97.26	97.37
	RankNet	96.67	96.37	96.63
	ListNet	96.66	96.66	96.66
	Random Forest	**97.49**	**97.63**	**97.59**
	LamdaMART	97.19	97.38	97.29
	Cordinate Ascent	96.18	96.18	96.18

C. Performance of MLC-Learning to rank

This scenario focuses on using learning to rank for multilabel classification. In this scenario, we employ combination of algorithm, feature, and term representation to find the right configuration of MLC-Learning to rank for this problem.

In this scenario, we employ learning to rank as a multilabel classification, and NDCG@5 for measuring the performance for each algorithm. We do not use threshold yet.

Different from Yang & Gopal, we do not consider the meta-level feature. We apply term representation, complaint topic, and existence label in text and all labels of complaint text.

The best performer of this scenario is LambdaMART with tf-idf terms and complaint topic as features.

TABLE III. THE EXPERIMENT RESULT OF THIRD SCENARIO

Algorithm	Binary	Count	TF-IDF
Vector of terms			
ListNet	40.29	40.29	37.82
MART	62.65	62.23	60.20
LamdaMART	62.55	62.82	**60.72**
Random Forest	**64.06**	**64.24**	59.60
Vector of terms and complaint topic			
ListNet	60.51	60.51	60.51
MART	65.04	65.43	65.51
LamdaMART	**70.27**	**69.52**	**70.69**
Random Forest	65.08	65.08	64.24
Vector of terms and existance label in text			
ListNet	60.51	60.51	60.51
MART	63.68	62.42	63.22
LamdaMART	**67.63**	**67.55**	**67.92**
Random Forest	62.52	62.32	61.87

D. Comparison between scenarios

For comparison, we need a discriminative measurement for all scenarios that will show performance in same measurement method. So, we have accuracy of two steps. First is accuracy for primary agencies and second is accuracy for secondary government agencies. Since secondary have more one label, we compute the subset accuracy which is used in multilabel classification.

First scenario use this performance measure. So, we do not have to convert it. For second scenario, we need to convert the performance measure. We apply multilabel and ranking result to convert it. For the third scenario, we apply the threshold method for the data result. After threshold, we can get the primary and secondary label that we will use to calculate the performance for comparison.

We take the best performance of each scenario in every alternative of feature vectors. With only vector of terms as feature, we apply Naïve Bayes and LP-SVM for scenario I; LP-SVM and MART for scenario II; and Random Forests for scenario III.

With vector of terms and topic of complaint as feature, we employ SVM and ML-kNN for scenario I; LP-SVM and Random Forests for scenario II; and LamdaMART for scenario III.

With vector of terms and existence label in text, we employ SVM and ML-kNN for scenario I; LP-Naive Bayes and Random Forests for scenario II; and LamdaMART for scenario III.

TABLE IV. COMPARISON

Scenario	Accuracy	
	Primary	Secondary
Vector of terms		
I	55.76	57.06
II	49.93	50.77
III	49.44	78.31
Vector of terms and complaint topic		
I	65.65	71.80
II	52.32	53.29
III	59.46	77.32
Vector of terms ,complaint topic, and existance label in text		
I	64.35	72.33
II	50.91	51.77
III	56.41	76.98

The best performance of prediction is belong to scenario I which is typical solution with vector of terms and topic of complaint as feature. But as we can see that secondary agencies prediction of scenario III which use learning to rank for multilabel classification is more accurate than scenario I. So, learning to rank for multilabel classification has problem with predicting first rank of categories which has only one category.

Otherwise, scenario II which use combination multilabel classification and learning to rank always get lower accuracy than another scenario. Since the multilabel classification has lower accuracy, learning to rank is affected with that (see TABLE II).

Fig 3 shows the example of complaint text to testing every scenario. Since vector of terms and complaint topic is the best feature set, we employ this feature set to this example.

> Jalan Jenderal Sudirman Gg. Manunggal 2C RT 06/01, Kelurahan Cijerah, Kecamatan Bandung Kulon terjadi longsor jembatan. Mohon bantuan nya terima kasih.

Figure 3. Example of complaint text in LAPOR

TABLE V. PREDICTION RESULT OF EXAMPLE

	Actual Label	Prediction Label		
		Scenario I	Scenario II	Scenario III
Primary	Dinas Bina Marga dan Pengairan	Dinas Bina Marga dan Pengairan	PD Kebersihan	Dinas Bina Marga dan Pengairan
Secondary	Dinas Perhubungan, Kecamatan Bandung Kulon, Dinas Sosial	Dinas Perhubungan, Kecamatan Bandung Kulon, Dinas Sosial	-	Satuan Polisi Pamong Praja, Disnas Perhubungan, Badan Pengelolaan Lingkungan Hidup

TABLE V shows the prediction results of example in Fig. 3. Scenario I (single label and multilabel classification) show the perfect prediction for this example. Scenario III (learning to rank for multilabel classification) has right prediction in primary agency, but only get one of three right prediction in secondary agencies. Otherwise, the scenario II (combination of multilabel classification and learning to rank) get the wrong prediction in primary agency, and can not predict the secondary agencies.

VII. CONSIDERATION

Our methods to employ learning to rank have been proposed and evaluated in this experiment. Unfortunately, these methods have not shown better performances compared to the typical approach. Combination approach of multilabel classification and learning to rank has not shown the good performance. Otherwise, the method that used MLC Learning to Rank has shown the better performance than the other. But it still can not outperform the typical solution of this problem.

For the combination method of multilabel classification and learning to rank, learning to rank has shown the good performance, but the multilabel classification has shown poor performance. Since the performance depends on both models, the multilabel classification performance should be improved.

Another reason is that our dataset is imbalanced dataset, so these methods performance has not shown the best performance. We believe this method will work well in the future by applying some methods for handling imbalanced dataset.

VIII. CONCLUSION AND FURTHER WORK

Learning to rank approach for this problem has been proposed. Experiment shows that our learning to rank approaches still below the typical solution in machine learning, but we believe that it is still promising and we can improve this method in the next future works, including bigger dataset, better feature set, and applying method for handling imbalanced dataset.

Learning to rank for multilabel classification has shown the better result than combination of multilabel classification and learning to rank. It has shown the better accuracy than the typical solution in secondary prediction. The results show that LamdaMART, which is listwise approach in learning to rank, is the best algorithm for classifying the primary agency and the secondary agencies.

Finally, we have shown that learning to rank can be used in this problem. But we still need to improve our method. We believe that our models can help the government to manage their complaint management.

REFERENCES

[1] F. Sebastiani, "Machine Learning in Automated Text Categorization," *ACM Computing Surveys, Vol. 34, No. 1, March 2002*, 2002.

[2] "LAPOR! - Layanan Aspirasi dan Pengaduan Online Rakyat," UKP-PPP, 2013. [Online]. Available: https://lapor.ukp.go.id/.

[3] UKP-PPP, "Tentang LAPOR!," UKP-PPP, 11 2013. [Online]. Available: https://lapor.ukp.go.id/lapor/tentang_lapor.

[4] H. Li, "A Short Introduction to Learning to Rank," *IEICE Transaction on Information and Systems,* 2011.

[5] H. Li, "Learning to rank for information retrieval and natural language processing," *Synthesis Lectures on Human,* 2011.

[6] E. A. Tanaka and J. Baranauskas, "An Adaptation of Binary Relevance for Multi-Label Classification applied to Functional Genomics," *Proceedings of the XXXII Congress of the Brazilian Computer Society, XII Workshop on Medical Informatics*, 2012.

[7] G. Tsoumakas and I. Kataki, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining,* vol. 3, no. 3, pp. 1-13, 2007.

[8] A. Shashua and A. Levin, "Ranking with large margin principle: Two approaches," in *Advances in Neural Information Processing Systems*, 2002.

[9] C. Burges, "Learning to Rank using Gradient Descent," *Proceedings of the 22nd International Conference on Machine Learning,* 2005.

[10] L. Breiman, "Random Forests," 2001.

[11] R. Herbrich, T. Graepel and K. Obermayer, "Large Margin Rank for Ordinal Regression," in *In Preceeding of Advances in Neural Information Systems*, 2000.

[12] C. J. Burges, R. Ragno and Q. V. Le, "Learning to Rank with Nonsmooth Cost Functions," in *NIPS*, 2006.

[13] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai and H. Li, "Learning to Rank: From Pairwise Approach to Listwise Approach," *ICML '07 Proceedings of the 24th international conference on Machine Learning,* 2007.

[14] Y. Xu and L. Chen, "Term-frequency based feature Selection methods for Text Categorization," *2010 Fourth International Conference on Genetic and Evolutionary Computing,* pp. 281-283, 2010.

[15] C. J. Burges, "From RankNet to LambdaRank to LambdaMART: An Overview," *Microsoft Research Technical Report MSR-TR-2010-82,* 2010.

[16] K. Jarvelin and J. Kekalainen, "IR evaluation methods for retrieving highly relevant documents," *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '00,* no. ACM, 2000.

[17] Y. Yang and S. Gopal, "Multilabel classification with meta-level features in a learning-to-rank framework," *Machine Learning Journal 2011*, pp. 47-68, 2011.

[18] K. Trohidis, G. Tsoumakas, G. Kalliris and I. Vlahavas, "Multilabel classification of music into emotions," *Proc. 9th international conference on music information retrieval (ISMIR 2008),* 2008.

[19] M. Boutell, J. Luo, X. Shen and C. Brown, "Learning multi-label scene classification," *Pattern Recognition, 37(9),* pp. 1757-1771, 2004.

[20] A. Elisseeff and J. Weston, "Kernel methods for multi-labelled classification and categorical regression," *Advances in neural information processing systems,* vol. 14, pp. 681-687.

[21] A. Ganapathiraju, J. Hamaker and J. Picone, "Support vector machines for speech recognition," *International conference on spoken language processing,* pp. 2923-2926.

[22] W. Cheng and E. Hüllermeier, "Combining instance-based learning and logistic regression for," *Machine Learning*, vol. 76(2–3), p. 211–225, 2009.

[23] M.-L. Zhang and Z.-H. Zhou, "A k-nearest neighbor based algorithm for multi-label classification," *Proceedings of the 1st IEEE International Conference on Granular Computing,* 2005.

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

[25] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. H. Witten, "The WEKA Data Mining Software: An Update," *SIGKDD Explorations*, vol. 11, no. 1, 2009.

[26] G. Tsoumakas, I. Katakis and I. Vlahavas, "Mining Multi-label Data," in *Data Mining and Knowledge Discovery Handbook, O. Maimon, L. Rokach*, Springer, 2010.

[27] V. Dang, "The Lemur Project - Wiki - RankLib," Lemur Project, [Online]. Available: http://sourceforge.net/p/lemur/wiki/RankLib/.