Journal of Engineering and Applied Sciences 12 (18): 4651-4656, 2017
ISSN: 1816-949X

Machine Learning-Based Topical Web Crawler: An Ensemble Approach Incorporating Meta-Features

Tae Jun Kim and Han-Joon Kim
School of Electrical and Computer Engineering, University of Seoul, 163 Seoulsiripdae-ro, Dongdaemun-Gu, 02504 Seoul, Korea

Abstract: A topical web crawler is to collect web pages that describe some pre-specified topics. The web pages collected by the topical crawler share the same or similar words and however among them not a few pages can be irrelevant to the given topics. In particular, the performance of topical crawler degrades for a more specific topic. To achieve successful topical crawling, an additional job is required to actively filter out the pages irrelevant to the given topics. For this we propose an ensemble-style machine learning architecture that can effectively handle not only literal term features but also numeric meta-features to improve topical web crawler; in our work we intend to more precisely crawl the web pages about 'fire accidents' as a specific topic. In case of the fire we have found that significant meta-features for topical crawling include the information of tags, the number of words in the title, the number of person names, the number of location names of web pages and so forth. For the numeric meta-features we use the logistic regression and random forest learning algorithms and for the literal word features, Naive Bayes and support vector learning algorithms. Through extensive experiments using the fire accident-related news articles we prove that the proposed method outperforms the conventional ones.

Key words: Machine learning, ensemble, web crawler, meta-features, filtering, extensive

INTRODUCTION

Topical web crawling refers to the technique of collecting web pages corresponding to information requests expressed by user queries or profiles of interest. (Yang, 2016). The topical crawler can help you extract a small but focused subset of web pages from the entire web and this collection can be mined with the appropriate information using text mining, indexing and ranking tools. However, as shown by Nugroho and Suryanegara (2016), Wang *et al.*(2009), it is not easy to retrieve only web pages relevant to pre-specified topics although, diverse approaches range from a simple keyword matching algorithm to complex machine learning classification algorithms.

A key motivation for topical web crawling is to populate the repository of vertical web search systems. Although, it is not necessarily aimed at constructing the whole web search system we need to gather documents related to a specific topic for various application needs such as text analysis. Still, many of currently developed topical crawlers do not have a low false positiveness rate since their approaches are basically dependent upon 'bag-of-words' document representation (Breiman, 2001). In this study, we propose a machine learning-based topical crawling technique that considers not only literal word features but also numeric metadata features; moreover, the proposed method focuses upon collecting only news articles relevant to fire accident. Figure 1

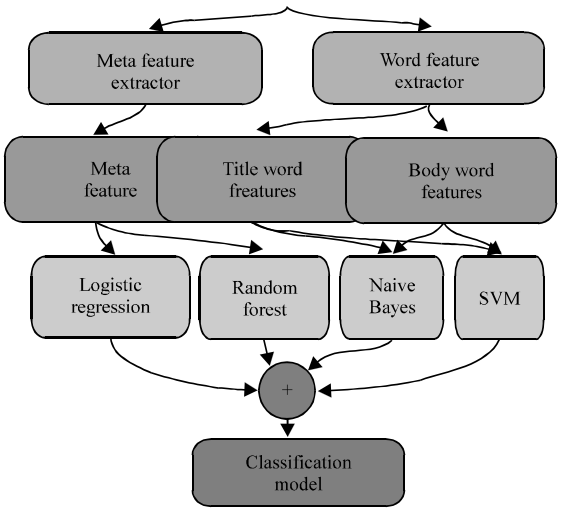

Fig. 1: The proposed ensemble-based machine learning architecture that fuses different meta-features

Corresponding Author: Tae Jun Kim, School of Electrical and Computer Engineering, University of Seoul, 163 Seoulsiripdae-ro, Dongdaemun-Gu, 02504 Seoul, Korea

shows our proposed machine learning architecture for fire related news study crawling. Briefly describing this architecture, several classification models are generated by adopting the machine learning algorithms (such as logistic regression, random forest, Naive Bayes and support vector machine) suitable for feature characteristics and then the models are combined with an ensemble technique, so as to improve the crawling (or filtering) performance. Since, topical crawling eventually deals with the filtering problem we will consider the terms 'crawling' and 'filtering' to have the same meaning throughout the study.

MATERIALS AND METHODS

Feature exploration: As mentioned earlier in this study, we focus upon using different types of metadata features of documents as additional features of classification models in order to improve the conventional filtering methods that use only the word features. Firstly, in this study we describe the overall exploration of metadata features to understand why metadata are important and helps in document filtering.

Metadata 1 (number of characters): As the first meta-feature, we introduce the number of characters occurring in the title and body of news articles. The scatter plot of this feature is shown in Fig. 2 with separating fire accident-related documents and non-fire accident documents. In this Fig. 2, we see that the fire accident related documents are distributed on the left side and non-fire documents are spread evenly in terms of the number of characters; fire accident-related documents are concentrated in places where the number of characters in the body text is small. Through this observation, the number of characters occurring in the title or body text is considered to be a good feature that can judge fire news articles well.

Metadata 2 (number of named types): Table 1 shows the average frequency of occurrences of words according to their named types and position. For instance, the average frequency of words in a person name in the title of a non-fire accident document is 0.13 and 0.04 in the case of a fire accident-related document that is the probability that the features of person name appear in the title of a non-fire accident document is about 2.94 times higher than that in a fire accident-related document. This indicates that the frequency of the named types and positions of words is also a good feature for fire accident-focused web crawling.

Table 1: The average frequency of words according to their named type and position

Named type	Position	Fire	Non-fire	Ratio (non-fire/fire)
Place name	Title	0.71	0.53	0.75
	Body	4.38	7.23	1.65
Person name	Title	0.04	0.13	2.94
	Body	1.91	3.92	2.05

Table 2: The ratio of syntactic tags

Tags	Position	Fire	Non-fire	Non-fire/fire
Foreign word	Title	0.0014	0.0103	7.15
Chinese word	Body	0.0001	0.0006	5.41
Designator	Title	0.0002	0.0007	3.90
Stem	Title	0.0011	0.0035	3.25
Stem	Body	0.0009	0.0024	2.83

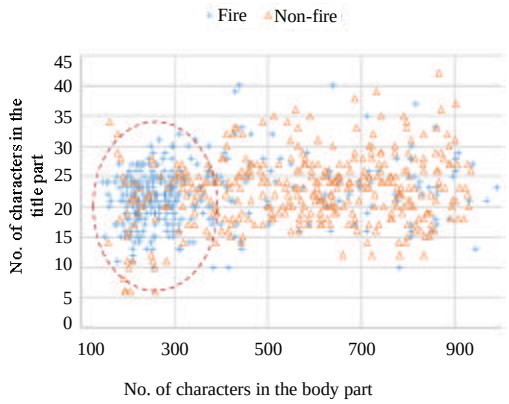

Fig. 2: The scatter plot of the number of characters occurring in the title and body

Metadata 3 (ratio of syntactic tags): In our research, a syntactic tag refers to a sytactic term assigned to each of significant words which includes part-of-speech (such as verb, stem, designator and foreign word) or the format type of word (such as number). Table 2 shows the ratio of top 5 syntactic tags that appear more frequently in non-fire articles. For example, the frequency of foreign words in the title of a non-fire related news study is 7 times greater than in a fire accident related study. As shown in the Table, the ratio of syntactic tags in fire and non-fire related articles is significantly different and thus the ratio of syntactic tags can be a good indicator for topical crawling.

Consideration of variation of the BOW Model: We do not ignore conventional features of literal words although their performance is relatively poor in classifying a set of documents with words of similar meaning. In this subsection we describe ways to enhance the Bag-of-Words (BOW) Model with the metadata features.

Table 3 shows the rankings in terms of the number of occurrences of the word 'factory' in the news article. Of all the significant words occurring in the entire document,

Table 3: The frequency ranking of the word 'factory'

	Rank (± difference)	
Scope	Fire	Non-fire
Entire document	14	182
Title	3 (+11)	562 (-380)
Body	19 (-5)	174 (+8)

Table 4: The components of the meta-feature V_{meta}

Components	No. of dimensions
Number of characters in the title and body text	2
Number of place names in the title and body text	2
Number of person names in the title and body text	2
Ratio of word tags in the title text	16
Ratio of word tags in the body text	16
Total	38

the word 'factory' ranks 14th in the fire accident-related document and 182 in the non-fire document in terms of word frequency. The difference is more apparent when the frequency ranking of words is limited to the title area. In the case of fire accident-related documents, the ranking in the title increased by 11th ranks to 3rd whereas for non-fire documents, the ranking decreased to 562. In contrast, the rankings in the body text were 19th and 174th in the fire and non-fire related documents, respectively with slightly reducing the difference. Through, the above observations we think that if the word 'factory' appears in the title of a news article, the news article is very likely to be related to the fire accident. If we extract some useful features with considering the position where the word appears as in the above example, more effective classification models for document filtering could be developed.

Feature engineering with metadata: In the previous study, we looked at the different types of metadata as features for machine learning. This study, describes how to combine these metadata with existing word features.

Meta-features V_{meta}: We define a vector V_{meta} composed of new meta-features with the components shown in Table 4 which include the metadata from the news articles discussed in the previous atudy.

Meta-bag-of-words features V_{mbow}: Let V_{title} and V_{body} be word frequency vectors composed of words appearing in the title text and the body text, respectively. With V_{title} and V_{body}, the conventional bag-of-words vector V_{bow} can be expressed as follows:

$$V_{bow} = V_{title} + V_{body} \quad (1)$$

Here, the symbol '+' denotes the operator that adds two vectors with the same number of dimensions; for example, assuming V_{title} is <1, 0, 1, 0> and V_{body} is <1, 1, 1, 1>, V_{bow} is <2, 1, 2, 1>.

Table 5: Experimental data distribution

Variables	Fire	Non-fire	Total	Ratio (%)
Learning	380	534	914	68.9
Testing	158	254	412	31.1
Total	538	788	1.326	
Ratio (%)	40.6	59.4		

Table 6: A confusion matrix of classification model C_i

	Prediction	
Model C_i	Fire	Non-fire
Actual; fire	True positive (tp)	False negative (fn)
Non-fire	False positive (fp)	True negative (tn)

In our research, we intend to distinguish the words in the title text and the words in the body text, even if they are the same words. This is because the words appearing in the title can contribute greatly to identifying the documents with a pre-specified topic as mentioned earlier. To define such a meta-bag-of-words vector V_{mbow}, two vectors are concatenated as follows:

$$V_{mbow} = V_{title} \cdot\cdot V_{body} \quad (2)$$

Here, the symbol '||' denotes the operator that concatenates two vectors with different or equal number of dimensions; taking the above example, V_{mbow} is <1, 0, 1, 0, 1, 1, 1, 1>. That is unlike V_{bow}, V_{mbow} has two separate dimensions for the same word by assigning dimensions differently depending on whether the word comes from the title or the body text.

Lastly, in order to generate classification models for document filtering we use the vector V_{total} that integrates the two vectors V_{meta} and V_{mbow} of the training documents:

$$V_{total} = V_{meta} \cdot\cdot V_{mbow} \quad (3)$$

Experiments

Empirical setup: The set of news articles used for this experiment was generated by downloading the search result from the keyword 'fire' in Korean search portal named 'daum' at http://daum.net. Of these, 1,326 were randomly selected which are then manually classified into two classes: fire accident news and non-fire news. To perform the learning process for document filtering, 68.9% of the total data were used for learning and the rest of them for testing. Detailed experimental data distribution is shown in Table 5.

Performance evaluation: In order to evaluate the classification model learned we have used accuracy, precision, recall and F_1-measure as performance metrics. Given a confusion matrix of the classification model C_i as shown Table 6, its Accuracy (A_i), Precision (P_i), Recall (R_i) and F_1-measure (F_i) are computed as follows:

$$A_i = \frac{tp+tn}{tp+tn+fp+fn}$$
$$P_i = \frac{tp}{tp+fp} \; Ri = \frac{tp}{tp+tn} \; F_i = 2\times\frac{P_i\times R_i}{P_i+R_i} \quad (4)$$

In these experiments, our goal is to isolate only fire accident related news articles among news articles. Thus, we must build up the model to maximize the precision by reducing false positives as much as possible. Of course, the recall should also be reasonably high, so that actual fire accident articles are included in the filtered results. In other words, it is important to maximize the F_1-measure that encompasses both of these metrics.

Learning algorithms: We have tested most machine learning algorithms to find out which algorithm is best for each of feature types. Through, extensive experiments we have found that logistic regression (Peng *et al*., 2002) and random forest (Breiman, 2001) is appropriate for the meta-features denoted as V_{meta} and Naive Bayes (McCallum and Nigam, 1998) and support vector machine (Joachims, 1998) are best for the features denoted as V_{bow}, V_{mbow} and V_{total}.

Logistic regression is a regression algorithm in which dependent variables (i.e., features) are categorical. To optimize the logistic regression model we used the OWL-QN (Andrew and Gao, 2007) algorithm where the regularized loss function J'(w; X, y) to be optimized is regularized as follows:

$$J'(w;X,y) = J(w;X,y)+\alpha\Omega(w) \quad (5)$$

$$\Omega(w) = \beta\|w\|_1 + \frac{1}{2}(1-\beta)\|w\|_2^2 \quad (6)$$

Where:

J (w; X, y) = The original loss function for the input variables
X = Their weights w and output variable Y
α = The regularization parameter that indicates the degree of regularization
Ω (w) = The regularization function that uses elastic-net method (Zou and Hastie, 2005)
β = The weighted value of L1 normalization for L2 normalization

Random forest is an ensemble-style learning method for decision trees which can solve the overfitting problem of the decision tree algorithm. Here, the number of Trees (T) and the maximum Depth (D) of each tree can be adjusted so as to optimize the performance.

Naive Bayes is a probabilistic learning method based on the Baye's theorem which ignores dependency between features to reduce the complexity of probability computation. For a given data X = <x_1, x_2, ..., x_n> it is classified into the class k with the highest probability value using the following Eq. 7:

$$y = \arg\max_k \Pr(c_k)\cdot\prod_{i=1}^{n}\Pr(x_i|c_k) \quad (7)$$

$$\Pr(x_i|c_k) = \frac{N_{k,i}+\gamma}{N_k+\gamma n} \quad (8)$$

Where:

$N_{k,i}$ = The number of times the feature
x_i = Observed in class
k and N_k = The number of times all the features are observed in class k

To prevent the value of Eq. 8 from becoming zero, the laplace smoothing (Yuan *et al*., 2012) is used in which γ is the parameter that determines the degree of smoothing.

Support Vector Machine (SVM) is a non-probabilistic learning method that constructs a hyperplane or a set of hyperplanes in a high dimensional space which shows excellent performance in text classification. For optimization in our work we used the stochastic gradient descent (Bottou, 2010) where its regularized loss function is the same as that used for the logistic regression given in Eq. 9 and 10.

Table 7: Specification of model design

Model types	Machine learning algorithm	Feature types
LR_{meta}	Logistic Regression	V_{meta}
RF_{meta}	Random Forest	V_{meta}
NB_{bow}	Naive Bayes	V_{bow}
NB_{meta}	Naive Bayes	V_{mbow}
NB_{total}	Naive Bayes	V_{total}
SVM_{bow}	Support Vector Machine	V_{bow}
SVM_{meta}	Support Vector Machine	V_{mbow}
SVM_{total}	Support Vector Machine	V_{total}

Model design: To be noted is that we have adopted the learning algorithm suitable for each of features described. The features denoted as V_{meta} are applied to logistic regression and random forest which are known to have good performance for continuous input variables. The features denoted as V_{mbow} are applied to naïve Bayes and SVM. For comparison with the proposed algorithm the model using V_{bow} was used as a baseline. Table 7 shows various classification models that can be derived according to learning algorithms and feature types.

Model tuning: In these experiments, we tried to create the optimal classification model for each of model types. To optimize the hyper-parameters of each model, we performed the grid search with 10-fold cross validation (Thornton *et al*., 2013). Table 8 shows agrid for the candidate values of hyper-parameters of the learning algorithms selected. After determining the optimal

Table 8: A Grid for the candidates of hyper-parameters

Algorithms	Hyper-parameters	Candidates
Logistic	α	0.001, 0.01, 0.1, 1, 10, 100
Regression	β	0, 0.5, 1
Random	T	120, 300, 500, 800
Forest	D	5, 8, 15, 25
Naive Bayes	γ	0, 0.1, 0.3, 0.65, 1
SVM	α	0, 0.01, 0.1, 1
	β	0, 0.2, 0.5, 0.8, 1

Table 9: The optimal values of hyper-parameters

Models	Hyper-parameters	Best values
LR_{meta}	α,β	0.1, 0
RF_{meta}	T, D	800, 25
NB_{bow}	γ	0.1
NB_{mbow}	γ	0.1
Nb_{total}	γ	0.1
SVM_{bow}	α, β	0.1, 0.2
SVM_{mbow}	α, β	0.1, 0
SVM_{total}	α, β	0.1, 0

Table 10: Specification of ensemble models

Model types	Model components	Ensemble
M1	LR_{meta}, RF_{meta}, SVM_{mbow}	Class voting
M2	RF_{meta}, NB_{mbow}, SVM_{mbow}	
P1	LR_{meta}, RF_{meta}, SVM_{mbow}	Probability voting
P2	RF_{meta}, NB_{mbow} , SVM_{mbow}	

hyper-parameters, each model was rebuilt using the entire training data. Table 9 shows the optimal hyper-parameter values selected for each model after performing grid search.

Model integration: There are two ways to integrate both properties of V_{meta} and V_{mbow}. One is to concatenate two vectors as mentioned in Eq. 3 and the other is to combine different models according to the ensemble technique.

The ensemble technique has two ways of majority voting (James, 1998) which include 'class voting' and 'probability voting'. The class voting is a method of determining the most predicted value among the predicted class values of the model as the final predicted class $(\hat{y})$ which is formally described as follows:

$$\hat{y} = \arg\max_j \sum_i d_{ij},\ d_{ij} = \begin{cases} 1, & \text{if } C_i(x) = j \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

Where d_{ij} is a value of 1 if the prediction class of the model C_i equals j or 0 otherwise. By comparison, probability voting is a way of taking a class with a higher probability by computing the sum of the predicted probabilities for each class of models. The final predicted class $(\hat{y})$ of the probability voting is calculated as:

$$\hat{y} = \arg\max_j \sum_i r_{ij} \quad (10)$$

where r_{ij} is the probability of the model C_i for the class j. Table 10 shows the ensemble models and their components according to the way of model integration.

Table 11: Performance comparison of classification models learned

Models (C_i)	Accuracy (A_i)	Precision (P_i)	Recall (R_i)	F_1 (F_i)
Nb_{bow}	0.922	0.858	0.956	0.904
SVM_{bow}	0.903	0.824	0.949	0.882
LRf	0.827	0.760	0.804	0.782
RF_{meta}	0.871	0.872	0.778	0.823
NB_{mbow}	0.922	0.856	0.955	0.903
NB_{total}	0.917	0.851	0.949	0.897
SVM_{meta}	0.932	0.886	0.942	0.913
SVM_{all}	0.919	0.913	0.872	0.892
M1	0.902	0.882	0.859	0.870
P1	0.934	0.939	0.885	0.911
M2	0.932	0.886	0.942	0.913
P2	0.924	0.861	0.955	0.906

RESULTS AND DISCUSSION

Table 11 shows the performance of classification (or filtering) models in terms of the evaluation metrics. The highest value for each metric is underlined. Firstly, let us look at the Rf_{meta} model using V_{meta}. Its accuracy and F_1-measure were 0.871 and 0.823, respectively although, the word frequency was not taken into account at all. Moreover, its precision is 0.872 that is higher than that of NB_{bow}. The support vector machine with the features of V_{meta} showed higher performance for all the metrics except the recall than that of the conventional bag-of-words models.

When comparing two NB models, 2 SVM models and four ensemble models, the performance of both P1 and P2 was better than that of the other models. Particularly, P1 has the highest precision of 0.939 which is an 8% improvement over NB_{bow}, the best conventional model. Here, its recall is 0.885 which is an acceptable level. In addition, its accuracy and F1-measure are 0.934 and 0.911, respectively both of which are 1% higher than NB_{bow}. The model with the highest performance in terms of F_1-measure is M2. These experimental results indicates that the ensemble models outperforms other models that are obtained through learning training data corresponding to the features of V_{total}.

CONCLUSION

In this study, we propose an ensemble-style machine learning architecture to isolate only documents related to fire accident to achieve true topical crawling. Its model parameters include various meta-features related to the position of words, their syntactic tags and their frequency. Through a large number of experiments, we found the machine learning algorithms appropriate for feature characteristics and the models learned are combined with an ensemble technique, so as to improve the crawling (or filtering) performance. In the future, we plan to generalize the proposed meta-features and the ensemble structure to identify documents about various topics including fire accident.

ACKNOWLEDGEMENT

This research was supported by a grant from Urban Architecture Research Program funded by Ministry of Land, Infrastructure and Transport of Korean government (No. 16AUDP-B100356-02).

REFERENCES

Andrew, G. and J. Gao, 2007. Scalable training of L 1-regularized log-linear models. Proceedings of the 24th International Conference on Machine Learning, June 20-24, 2007, ACM, Corvalis, Oregon, USA., ISBN:978-1-59593-793-3, pp: 33-40.

Bottou, L., 2010. Large-scale machine learning with stochastic gradient descent. Proceedings of the 19th International Conference on Computational Statistics (COMPSTAT'10), August 22-27, 2010, Springer, Paris France, pp: 177-186.

Breiman, L., 2001. Random forests. Mach. Learn., 45: 5-32.

James, G., 1998. Majority vote classifiers: Theory and applications. Ph.d Thesis, Stanford University, Stanford, California.

Joachims, T., 1998. Text categorization with support vector machines: Learning with many relevant features. Proceedings of the 10th European Conference on Machine Learning, Chemnitz, Germany, April 21-23, 1998, Springer, Berlin, Heidelberg, pp: 137-142.

McCallum, A. and K. Nigam, 1998. A comparison of event models for naive bayes text classification. Proceedings of the Workshop on Learning for Text Categorization, July 26-27, 1998, AAAI Press, Madison, Wisconsin, USA., pp: 41-48.

Nugroho, M.S. and M. Suryanegara, 2016. Analysis of interference of Unmanned Aircraft System (UAS) and ixed service at frequency band 12.5-12.75 GHz by considering the factor of rain attenuation. J. Adv. Technol. Eng. Stud., 2: 164-169.

Peng, C.Y., K.L. Lee and G.M. Ingersoll, 2002. An introduction to logistic regression analysis and reporting. J. Educ. Res., 96: 3-14.

Thornton, C., F. Hutter, H.H. Hoos and K.L. Brown, 2013. Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 11-14, 2013, ACM, Chicago, Illinois, ISBN:978-1-4503-2174-7, pp: 847-855.

Wang, C., Z.Y. Guan, C. Chen, J.J. Bu and J.F. Wang *et al*., 2009. On-line topical importance estimation: An effective focused crawling algorithm combining link and content analysis. J. Zhejiang Univ. Sci. A., 10: 1114-1124.

Yang, F.J., 2016. The user interface design of an intelligent tutoring system for relational database schema normalization. Intl. J. Technol. Eng. Stud., 2: 70-75.

Yuan, Q., G. Cong and N.M. Thalmann, 2012. Enhancing naive bayes with various smoothing methods for short text classification. Proceedings of the 21st International Conference on World Wide Web, April 16-20, 2012, ACM, Lyon, France, ISBN:978-1-4503-1230-1, pp: 645-646.

Zou, H. and T. Hastie, 2005. Regularization and variable selection via the elastic net. J. Royal Stat. Soc., 67: 301-320.