Detecting Latent Training Needs in MOOCs Forum by Using Stack Exchange as Training Data

Fu-Yin Cherng, Pierre Dillenbourg, Robert West

Ecole Polytechnique Fédérale de Lausanne, Switzerland

{*fu-yin.cherng,pierre.dillenbourg,robert.west*}*@epfl.ch*

ABSTRACT

Detecting training needs can assist school and cooperate association to make accurate decisions when constructing new training materials and curriculums. However, the process of identifying the topics of training needs and training offer establishments often cost amount of time. Thus, in this paper, we develop a workflow to facilitate the processing of training requires detection by using the technology of data mining. We used the tags of questions in Stack Exchange of Signal Processing as the potential training topics in the field of signal processing. Next, we built the machine learning model to predict the tags on testing question posts. To tested the generalization of our model, we applied the model to the unlabeled posts from MOOC forum of Digital Signal Processing course. Based on the results of predictions on these two datasets, we provide the primary evidence for the feasibility of detecting the topics of training needs in the fashion of digital epidemiology.

Author Keywords

Digital Education, Data Mining, Natural Language Processing

INTRODUCTION

In the era of digital information and big data, using data mining to understand the trend of the word or the customers are widely used in several fields and industries. Previous studies predicted heart disease mortality or AIDS treatments by analyzing people's tweets [3, 5]. Furthermore, Agrawal et al. proposed an approach to detect learner's confusion in MOOC forum and based on the content of confusion; their method can recommend related video clips to the learners [1]. In the commercial applications, by analyzing and predict the potential user interests and product review, the companies like Amazon and Spotify can provide or recommend several options to accurately meet their customers' needs.

Therefore, we considered that this idea of digital epidemiology could be extended to detect the training needs of a vocational domain or online learning. Because traditional methods to identify training needs are often conducted by collecting self-report survey or meetings of corporate associations, these methods usually cost a significant amount of the time to get the final result, and then another few more years are spent to establish the new training offers. However, judging from the previous examples of applied data mining, we believed that by collecting information and data relevant to training needs, we could find out the topics inside these digital traces and facilitate us to identify the training needs.

Our goal is to detect the latent training needs from digital traces. First, we need to decide is that which digital traces we should select, because several data sources might contain information about training needs, such as JOBS forum in Reddit, the description of job offers and Q&A forum. In this work, we chose the popular Q&A forum: Stack Exchange as our digital traces, because the content of question posts can directly reflect the topics that the people in this field have the questions with and want to learn about.

Tags are usually treated as a concise indication of the certain semantic aspect of related items [8]. Therefore, in this work, we reframe our goal of detecting training needs into detecting the tags of question posts. Without a doubt, there is other information which can also be used as the representation of the topics of training needs. But we considered the tags of question posts have the most direct connection with the topic of training needs and also a good start point; we focused on mining the tags of question posts in this work. Moreover, we want our method to be generalizable to other data source of the same domain. Training machine learning model required large dataset. However, not all the data source is large enough to build their machine learning models. If our method can be used in a various data sources, we can aggregate the detected topics from this source and obtained the overall and general training needs which exist in the different data sources. For example, we can apply the model trained by data of Stack Overflow to detect the topics of questions in other programming discussion forums. The other data source we choose in this work is the discussion forum of MOOC course "Digital Signal Processing", and for the data source used as training data, we selected the Stack Exchange of Signal Processing. We used the tags and posts from Stack Exchange of Signal Processing to build the classifier, and tested our classifier by applying it on the testing data from Stack Exchange and MOOC, separately.

Our result shows that we can accurately detect the tags of testing data from the same data source as training data, and we found that the classifiers trained by different features could complement to each other to gain the better result. As for the classification result of testing data from different data source, although we still observe some reasonable tagging result, the deeper investigation and systematic evaluations are required in the future.

This work is intended to make three contributions:

- We develop multi-label classifiers to detect the potential tags automatically from the posts of Stack Exchange. We

also compared the classification results from the classifiers training by different features.

- We demonstrated the workflow of how to train classifier of tags from one dataset and applied to another data source in the same domain.
- Based on our results, we revealed that it is feasible to extract latent topics of training needs from several related digital traces.

RELATED WORK

Since tags are usually used as a navigation among a massive data set, and the useful tools for users to find the information they want, it is common that the platform asks users to select some tags when they want to post in the forum. However, deciding which tags should be utilized often required knowledge and familiarity to this domain, and cost users times and effort to tag their post. Therefore, several studies are aiming to implement different kinds of tags recommendation system to facilitate the process of tagging.

Wang et al. used the information of historical tag assignment and the labeled LDA method to generate a list of recommending tags to users, and they reported they successfully improved the recommendation result comparing with the state of art method [9]. Song et al. developed the tag recommendation system for general context by combining the supervised and unsupervised methods. They also generated the network of tags based on the co-occurrence between each tag to help them build the recommendation system[7].

According to the related works mentioned above, we considered that using machine learning method and historical tag assignment to predict tags for un-label posts is promising. By mining the relationship between tags, we can use the topics of these tags as one indication for the topics of latent training needs.

GOAL AND RESEARCH QUESTIONS

Our goal is to develop a workflow that we utilized question posts of Stack Exchange as training data to build the machine learning models and predicted the tags of the testing posts from Stack Exchange and MOOC forum by using the models. The wider aim of this work is that by analyzing the tags that given to unlabeled posts, we want to extract the topics from these question posts, which can reveal the topics of latent training needs. To achieve this, we plan to investigate the following research questions.

RQ1: How similar the posts of Stack Exchange and MOOC forum are?
Although we selected the MOOC and Stack Exchange of the same domain, Digital Signal Processing (DSP), we still need to show that there is certain similarity between this two datasets to ensure that we can use the classifier trained by the posts of Stack Exchange to predict the tags for posts from MOOC forum.

RQ2: How well can we predict the tags for posts of Stack Exchange and MOOC forum?
The performance of our models has direct influence on whether we can accurately extract correct topics from posts. Therefore, we evaluate the quality of our models on Stack Exchange data by F1 score, specificity and Kappa value of prediction result. As for posts of MOOC forum, because we don't have the ground truth, we compared the tags generated by our model with the tags generated by Open Calais™[1].

DATA DESCRIPTION

Data Collection

We collected the posts from two resources: Stack Exchange of Signal Processing and course "Digital Signal Processing" from Coursera of cole Polytechnique Fdrale de Lausanne (EPFL). For the data of Stack Exchange of Signal Processing, we extracted the text contents of question posts, the tags of the post, and the list of all tag used in this stack from Stack Exchange Data Dump [2]. As for the data of MOOC forum, we extracted the text contents of all posts posted in the discussion forum of course "Digital Signal Processing." After filtering the posts without content and tags, we finally obtained 10,105 posts of stack exchange and the period of these posts is from 08/2011 to 09/2016, and 5,910 posts of MOOC forum and the period of these posts is from 02/2013 to 03/2015. There are totally 385 tags for stack exchange, and there are no tags for the posts of MOOC forum.

Data Processing

We take the bag-of-words approach in representing the posts of Stack Exchange and MOOC forum. First, we removed the URL, numbers, HTML hashtags, stop words and punctuations in posts. Second, after tokenizing the words, we stemmed the word to aggregate the words like "run" and "running." Finally, we adopted lemmatization to convert some words into their common base form of words (e..g., convert "am","are," "is" to "be").

Similarity between Stack Exchange and MOOC forum

To primarily analyze the similarity between posts of Stack Exchange and MOOC forum, we applied Latent Dirichlet Allocation (LDA) to all the posts from these two datasets. By using LDA provided from Gensim library, we obtained 20 topics for each dataset with ten topic words along with their probability for each topic. By inferring subjectively and manually, We found that there are four common topics among the two datasets (see Table 1). Therefore, we considered that at least for these topics, there is certain level of similarity existing between two datasets. However, deeper and systematic analysis are required, and we described it in the section of future works.

Furthermore, by observing the extracted topics of MOOC forum posts, we found that there are some noteworthy characteristics, which are different from the posts of Stack Exchange. The first is that the form of MOOC posts usually contained name and writing like an email, like "*Hi Nicholas Sorry but we have no control over the subtitles production Best Lionel.*" Another difference is that most of the posts content are related to homework or exam of the course, and

[1] http://www.opencalais.com/
[2] https://archive.org/details/stackexchange

sometimes the learners directly used the number of question and homework to represent the concepts, for example "*I think I'm stuck with the same one! HW1 Question 4?*" Generally speaking, because the MOOC forum is not only used for asking questions, besides domain-related topics and questions, MOOC posts tend to contain the topics related to announcements of the course, the reports of errors in videos and slides and so on.

Although there are some differences between the posts of Stack Exchange and MOOC forum, we did find some questions which are similar with questions in Stack Exchange, and there are also some common keywords in both datasets (e.g., "frequency", "image" and "filter"). If we can extract the domain- and course-related questions from the post of MOOC forums, we believed that we could analyze the similarity between these two datasets accurately.

OVERALL ARCHITECTURE

Figure 1 illustrated the overall architecture of this work. We first take a set of Stack Exchange posts as input, processing them by the methods mentioned in Data processing section. After we obtained the bag-of-words of each Stack Exchange post, we put them into the step of features extraction to gain the feature vectors that represent each post. The detail methods of feature extraction are described in the next section.

We used 80% of all Stack Exchange post as training data and used the remaining 20% as testing data. Because each post usually contained more than one tag, we conducted multi-label classification by generating multiple binary classifiers to predict whether the input post has the certain tag or not and then aggregating their predictions to obtain final tag list of the input post [6]. Each binary classifier is corresponding one tag that we selected from 385 tags used by Stack Exchange of Signal Processing. The basic idea of tag selection is to extract the most popular and important tags according to how many proportions each tag have because we want to know the important topics of training needs instead of those minor topics. We explain how we select tags in Tags Selection section.

After we had built the classifiers and testing with Stack Exchange posts, we applied these classifiers to MOOC forum posts to see whether we can give the MOOC posts sensible list of tags.

FEATURE EXTRACTION

We extracted two kinds of features to represent our datasets: popular words counts and 100-dimensions vector from Word2Vec.

The idea of using counts of popular words as features is that we believe that for most of the tags, there would be a corresponding set of words with high co-occurrence rate with the tag. We considered the counts of these words in each post can be used as the strong predictors to tell us which tags belong to this post. To construct the feature of popular words counts, we got the list of ten words that have the highest frequency of each tag. We referred these words as popular words for each tag. Then, we calculated the counts of these words for each

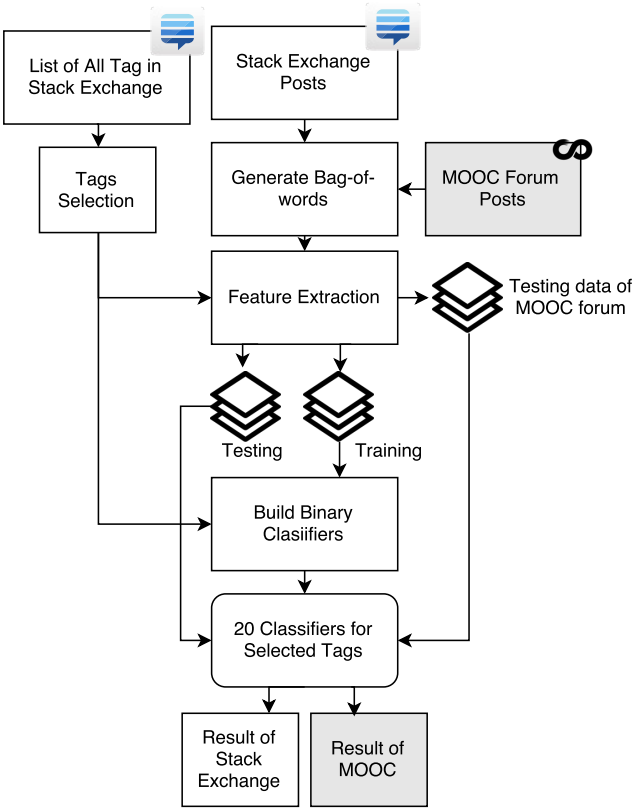

Figure 1. The workflow of the overall architecuture of our method.

post in training data and used these number of count as the features of popular words counts.

Wrod2Vec is a popular word embedding model which use vector space to represent each word [4]. The previous study reported that using word2vec to represent text data can achieve a better result than using count-based methods [2]. Therefore, we applied Word2Vec in this work, and we trained Word2Vec model by using all the posts and their tags as training data. We obtained a model that can provide us a vector with 100-dimensions for each word existing in training data. Next, by averaging the vectors of all the word in the post, we obtained the 100-dimensions vector for the post. Then, we took the number of each dimension as a feature, so we obtained 100 features for each word.

TAGS SELECTION

There are totally 385 tags used in Stack Exchange of Signal Processing. However, we found that note all tag is evenly used. There are even some tags used only once. Previous work also mentioned tags is noise because users would express the same concept by using various words [8].

To reduce the number of tags, we conducted two-step to filter out those minor tags. First, we descendingly sorted the tags based on their percentage of the count. Then we added the percentage of counts from the first tags tile the sum of count percentage nearly reach 80%. By the first filer, we obtained 93 tags out of 385 tags.

Table 1. The table of common topics generated by applying Latent Dirichlet allocation. The names to each common topic are derived from their topic words. For each common topic of one dataset, we listed five topic words sorted by their probability.

Common Topic: Code	**Topic Word (Probability)**
Stack Exchange	double (0.044), int (0.038), output (0.029), float (0.024), code (0.019)
MOOC Forum	input (0.036), example (0.023), output (0.023), line (0.019), code (0.018)
Common Topic: Filter	
Stack Exchange	filter (0.04), state (0.024), kalman (0.022), system (0.02), delay (0.017)
MOOC Forum	slide (0.032), filter (0.019), coefficient (0.018), operation (0.016), diagram (0.014)
Common Topic: Frequency	
Stack Exchange	signal (0.122), noise (0.047), frequency (0.022), power (0.012), sample (0.009)
MOOC Forum	signal (0.047), frequency (0.035), time (0.019), sample (0.017), would (0.013)
Common Topic: Image	
Stack Exchange	image (0.085), value (0.015), kernel (0.014), camera (0.013), gaussian,(0.013)
MOOC Forum	value (0.048), amp (0.029), image (0.024), end (0.02), picture (0.018)

Second, by plotting the percentage of counts of all tags (see Figure 2), we found that there is a sharp decline in the index of 20 in the curve of count percentage, which means that the important tags lie from the index 1 to 20. Therefore, we selected the tags from the index 1 to 20 as the target tags we want to predict. Table 2 shows the information of the tags we selected.

To know whether these tags belong to similar topics, we conducted clustering to know how these tags related to each other semantically. We used the Word2Vec model we built to generate vectors for each tag. By calculating the distance between each vector, we can construct a distance matrix which reveals how each tag semantically related to each other. We utilized k-means (k=7) to do clustering on the tags. From Figure 3, we can found that there are seven clusters, and we listed the tags and their group number in Table 3. From the clustering result, we can at least make sure that not all tag indicates the same or similar topics.

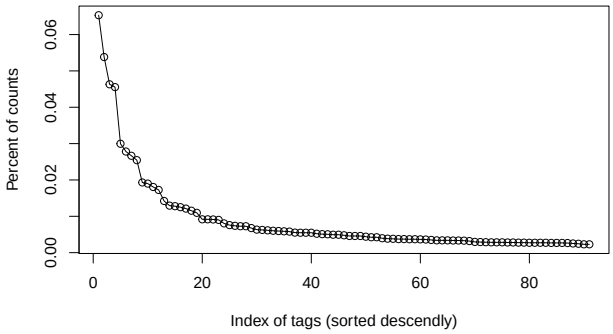

Figure 2. The percentage of counts for the tags.

APPLY MULTI-LABEL CLASSIFIER ON POSTS OF STACK EXCHANGE

We used Random Forests to build 20 binary classifiers corresponding to 20 target tags. Because the number of posts without this tag is always larger than the number of post with this tags, we adopted ROSE package in R to do oversampling on training data when we trained classifier of each tag. In this

Table 2. The list of selected tags and their number and percentage of counts.

Tag Name	**Count**	**% of Count**
image-processing	1681	6.5%
matlab	1385	5.4%
fft	1192	4.6%
filters	1172	4.6%
discrete-signals	771	3.0%
signal-analysis	716	2.8%
audio	686	2.7%
fourier-transform	655	2.5%
computer-vision	497	1.9%
frequency-spectrum	489	1.9%
filter-design	465	1.8%
noise	444	1.7%
sampling	366	1.4%
frequency	333	1.3%
dft	329	1.3%
digital-communications	322	1.3%
image	312	1.2%
convolution	297	1.2%
wavelet	282	1.1%
algorithms	236	0.9%

Table 3. The cluster number for each group of tags which grouping by applying k-means clustering.

Cluster	**Tag Name**
1	audio
2	frequency, image, noise
3	convolution, dft, fft, matlab, sampling
4	discrete-signals, filter-design, signal-analysis, fourier-transform, frequency-spectrum, filters
5	wavelet
6	digital-communications, algorithms
7	computer-vision, image-processing

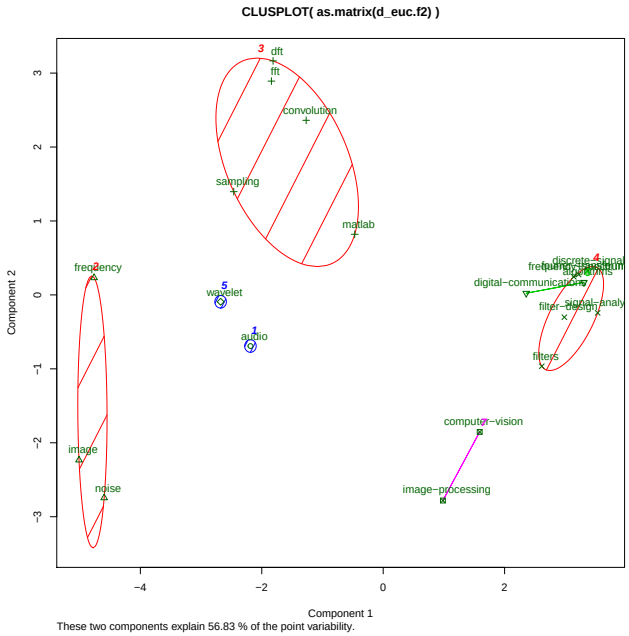

Figure 3. The result of cluster. The tags contain in cluster 4 is

way, we can avoid the misleading performance of the classifiers caused by imbalance data category.

By applying these classifiers on the test data of Stack Exchange, we used F1 score, specificity (true negative rate; positive class stands for prediction of not having this tag) to measure the quality of our classifier of each tag, and used Kappa value to measure the stability of our classifiers. In the following section, in addition to report the result, we also compare the difference between classifiers trained by features of popular word counts and Word2Vec.

Result and Discussion

Figure 4 shows the performance of our classifier of each tag by using word count as training features. We found that for most of the classifier, their F1 scores are higher than 90%, except for classifier of "filter-design". If we look the value of specificity, we can see more than half of all tag have specificity below 50%, which means that they ignored more than half of the posts that should be labeled with these tags.

Figure 5 shows the classification result of the classifiers trained by using Word2Vec features. We can see that the value of F1 score and specificity of the classifiers trained by Word2Vec features are similar. This result indicates that these classifiers can correctly tag the posts as many as possible.

To know whether there are significant differences between the performance of the classifiers trained by word counts and Word2Vec features, we applied t-test to F1 score, specificity and Kappa of this two classifier. In the following section, we used *word-count classifier* to stand for the classifier trained by features of word counts, and used *Word2Vec classifier* to stand for the classifier trained by features of Word2Vec.

We found that average F1 score of word-count classifier are significantly higher than Word2Vec classifier ($F(1,1)$=23.7, p<.0001), while Word2Vec classifier has higher specificity ($F(1,1)$=36.9, p<.0001). This result indicates that word-count classifier give tags to the post strictly, but if the tags were given, usually the tags can accurately represent the topic of the post. On the other hand, Word2Vec classifier is better at finding all tag that the post would have. Therefore, we can see from Table 4 word-count classifier only give the post one tags, and the Word2Vec give the post 11 tags.

As for the comparison of stability, we found that Kappa value of word-count classifier is higher than Word2Vec classifier ($F(1,1)$=10.7, p=0.0023), which means the word-count classifier can produce the consist quality of prediction repeatedly.

APPLY MULTI-LABEL CLASSIFIER TO POSTS OF MOOCS FORUM

We used the same word-count and Word2Vec classifier to predict tags to the posts of MOOC forum. Because MOOC posts weren't labeled originally, we evaluated our classifier by comparing the tags list from our classifiers and Open Calais™.

From the first post in Table 5, we found that tags contain "fourier" are given by both Open Calais™ and Word2Vec classifier. Also, tag "convolution" is given by our classifiers and Open Calais™ to the second post. We considered the reason of this fining might be that the word "convolution" appears four times in the second post and "convolution" itself is a strong predictor to the tag "convolution." If the length of the posts is too short, like the third post, there is no sufficient information to predict tag accurately. Nevertheless, we can see that because of the appearance of words like "unbounded, impulse, response", Word2Vec classifier and Open Calais™ assign the tags related to "signal" to the third post.

As we mentioned that there are various posts in MOOC forum, the further cleaning to MOOC posts is necessary. For example, replacing the words like "HW1" or "Q2" with the corresponding contents of the homework and question. Moreover, we can recruit the coders from the domain of signal processing to label tags to the posts manually, and take human-labeled tags as our ground truth to evaluate the performance of our classifiers on posts of MOOCs forum.

General Discussion and Findings

The goal of this work is to develop an approach to detect the training needs from Q&A forum (i.e., Stack Exchange and MOOC forum) by predicting the tags for the unlabeled posts. To show that we are able to achieve this goal, we intend to answer the two research questions.

First, we want to answer how well we can predict the tags for the posts of Stack Exchange and MOOC forum by our classifiers (RQ1). Based on our classification result on the posts of Stack Exchange, we show that both word-count and Word2Vec classifier can reach the mean F1 score above 80%, in which word-count classifier obtained significantly higher F1 score than Word2Vec classifier. This result indicates that

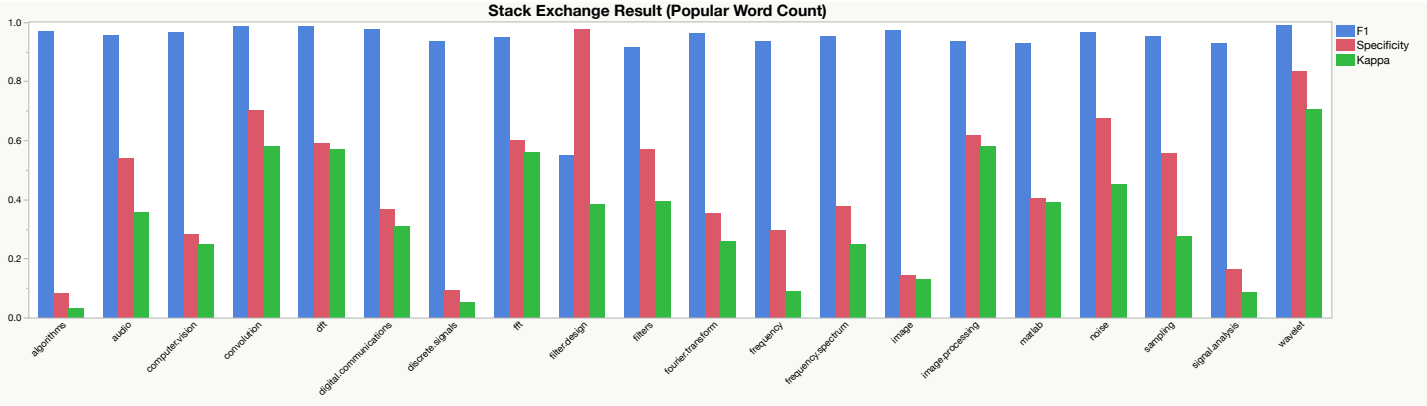

Figure 4. The result of classification on the testing posts of Stack Exchange using the counts of popular words as the training features.

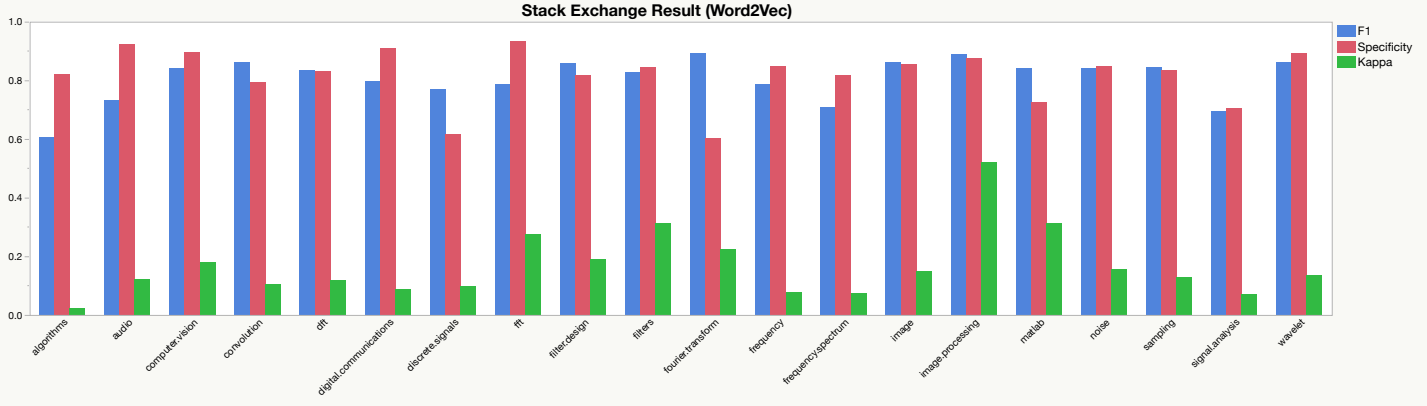

Figure 5. The result of classification on the testing posts of Stack Exchange using the vectors generated by Word2Vec model as the training features.

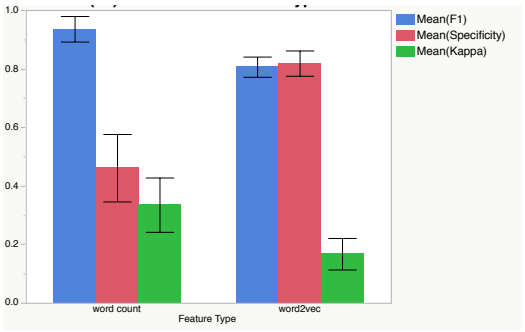

Figure 6. The comparison of the performance of classification between Word2Vec and word-count classifier.

word-count classifier can predict tags more accurately. However, the mean specificity value of word-count classifier is low which means it missed a lot of posts that should be predicted to have this tag. On the other hand, Word2Vec classifier is more stable than word-count classifier, and the mean value of specificity is also higher. Therefore, we considered that these two classifiers could complement each other. By combining the characteristics of word-count and Word2Vec classifiers, we can obtain the classification result with both high accuracy and specificity. Furthermore, this finding also motivates us to extract more kinds of features from the text content of the posts to improve the performance of our method.

By observing the performance of classification on the posts of Stack Exchange data, we can see that there are variations in all measurements. For example, in Figure 4, the classifier for tag "filer-design" has extremely high specificity value and low F1 score, which is opposite with the condition of other tags. We plan to conduct more experiments to investigate what factors of tags would influence their performance of classification.

To know whether it is possible to apply the classifier to the testing from difference dataset, another research question we want to whether the posts from Stack Exchange and MOOC forum are homogeneous (RQ2). There are several ways to measure and identify the similarity between two text datasets. In this work, we chose LDA to gain a general flavor of what are the overall topics inside these two datasets. By comparing the result of LDA on all the posts of Stack Exchange and MOOC forum, we extracted four common topics, which is "Code", "Filter," "Frequency" and "Image." Interestingly, we also found the corresponding tags in the list of Table 2. This finding reveals that in both posts of Stack Exchange and MOOC forum, the hot topics are similar.

Table 4. The example of our classification result on the posts of Stack Exchange and the ground truth.

Question Content	Original Tags	Predicted Tags (word2vec)	Predicted Tags (word count)
I've recorded a 2-sec pronunciation of a vowel sound. The first 0.12 or so seconds of the signal are shown below. Now, I've constructed an auto-regressive (AR) 8th-order model to compress this signal. (Actually, I'm just modelling 160 samples or 0.02 sec at a time.) The function in Matlab?s; System Identification Toolbox can estimate the parameters for an spectrum fit. My problem is choosing the stochastic input to the model filter. I suppose there's something better than white noise. The periodicity (14 periods per 0.02 seconds) leads me to think that an impulse train with the same period would be suitable. If so, how would I choose the amplitude, and how would I find the periodicity? ACF and PSD estimations are quite noisy. Am I even on the right track? image description here	digital-communications, autoregressive-model, speech	fft, filters, discrete-signal, signal-analysis, audio, frequency-spectrum, noise, sampling, frequency, digital-communications, algorithm	audio

Table 5. The examples of the classification results of our method and Open Calais^{TM} on the posts of MOOC forum.

Question Content	Open Calais (Social Tags)	Predicted Tags (word2vec)	Predicted Tags (word count)
Thanks for the help so far! Heres the last bit tripping me up still. I get that the delta function is symmetric. My confusion is in replacing the delta function in equation 2.18 with the impulse response function to yield equation 5.3 <formula> Correct me if Im mistaken, but Equation 5.3 is NOT valid if we used <formula> in place of <formula>. Thus, I'm confused because the apparent symmetry in equation 2.18 gets broken when going from 2.18 to 5.3, as in general we cannot assume the impulse response will be symmetric	Mathematical analysis, Digital signal processing, Generalized functions, Mathematics, Signal processing, Dirac delta function, Fourier analysis, Measure theory, Nyquist ISI criterion, Z-transform	discrete-signals, fourier-trnsform, filter-design, convolution	filters
The note says that the length of the resulting signal <formula> is: <formula> where: x is an input of length M, and h is an impulse response of size L. So we can rewrite this expression as: <formula> According to this length formula if <formula> then the size of the convolution is zero. Immediately before this formula, in the Linear Time-Invariant Systems note, there are two formulas for the convolution <formula> : one for the case <formula>, and one for the case, <formula>. It is clear that for <formula>, we simply have to use the corresponding formula. The questions are: a) There is no formula for <formula>, but the length of the convolution can be equal to <formula>. Is there typo or missing formula for <formula>? b) Should we use the formula <formula> for <formula>? c) In this course, when referring to convolution: should we restrict to <formula>, or should we use the formula for <formula> for all <formula>?	Mathematical analysis, Fourier analysis, Mathematics, Algebra, Digital signal processing, Signal processing, Overlap?add method, Functional analysis, Image processing, Linear time-invariant theory, Multidimensional discrete convolution, Convolution	matlab, fft, discrete-signal, frequency-specturm, sampling, dft, digital-communications, convolution	convolution
Is it possible that a system has an unbounded impulse response, i.e- <formula> but a bounded input to such system yield a bounded output?	Digital signal processing, Engineering, Control theory,Signal processing, Mathematical analysis, BIBO stability, Fourier analysis, Stability theory, Impulse response, Linear time-invariant theory, Systems theory	discrete-signals, signal-analysis, filter-design, convolution	filters

Additionally, we considered that the similarity between these two datasets could also be defined by judging the classification result directly. If the classifiers trained by the feature extracted from the posts of Stack Exchange can correctly predict the tags for the posts of MOOC forum, we could say that the posts of this two dataset are similar from another perspective. Therefore, we will investigate other methods or quantitative measurements for the similarity of different text datasets.

To sum up, we obtained accurate classification results (mean F1 score of two classifiers of all tag = 88%, SD=0.09) on the posts of Stack Exchange. We also observed some MOOC posts were assigned the tags that are semantically related to the text contents of the posts by our classifiers. Although we need more evaluation to determine whether our classification results on MOOC posts are correct or not, we collected the posts that are successfully tagged from MOOC posts and analysis why these posts can be tagged correctly by the classifiers trained by the posts of Stack Exchange. We provided a good start point to continuously develop and improve our approach to detect the training needs from various digital traces in a more comprehensive manner.

LIMITATIONS AND FUTURE WORKS

This work has several limitations should be noted. First, we need more evidence to support that we can use the classifiers trained by one dataset to predict the tags of other datasets. We can prove this concept by directly judging the result of classification or analysis documents similarity from other aspects.

Second, we selected important tags only based on their usage rate. For example, although the tag "frequency" is one of the tags we selected, however, there are also lots of materials to find the answer for the questions about frequency because this is a huge topic in signal processing. Therefore, in the future, we would like to integrate other criteria to represent the importance of the tags. For example, if we cannot find the MOOC course which description contained this tag, we can say that the training needs for this tag are more important than those tags which already have a bunch of tutorial on the Internet.

Third, we did not fully utilize all the clues hidden inside the text data of posts. There are other useful features we can use to train classifiers, including TF-IDF of words, co-occurrence rate between words and other contextual data (e.g., date and

users profiles). Furthermore, in this work we only try one machine learning method, we would like to compare the results of other machine learning method to see which methods are the best fits for our problem.

Finally, we will evaluate the classification result of MOOC post by comparing the results we obtained from crowdsourcing platform or labeled by experts of the same domain.

CONCLUSION

The overarching goal of this study was to develop an approach to detect the training needs to facilitate the overall progress of offer new training resources. By processing the text data from the posts of Stack Exchange, we built the classifiers to predict the tags for new coming posts from Stack Exchange and MOOC forum. We believed that the topics of the tags of the question posts could be treated as one of the indicators of training needs. In the classification results on the posts of Stack Exchange, our method high F1 score in average (88%), and we believe we can obtain better performance by combining different kinds of features. As for the classification results on the posts of MOOC forum, we considered that more evaluations are needed, but at the same time, we observed some promising examples that our method assigned the tags semantically related to the post contents. Taken together, our methods and results revealed that we could assign the correct tags to the new posts from the same data source successfully, and there is possibility to generalize our method to another kind of dataset in the same domain.

REFERENCES

1. Agrawal, A., Venkatraman, J., Leonard, S., and Paepcke, A. Youedu: addressing confusion in mooc discussion forums by recommending instructional video clips.
2. Baroni, M., Dinu, G., and Kruszewski, G. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *ACL (1)* (2014), 238–247.
3. Eichstaedt, J. C., Schwartz, H. A., Kern, M. L., Park, G., Labarthe, D. R., Merchant, R. M., Jha, S., Agrawal, M., Dziurzynski, L. A., Sap, M., et al. Psychological language on twitter predicts county-level heart disease mortality. *Psychological science 26*, 2 (2015), 159–169.
4. Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).
5. Salathe, M., Bengtsson, L., Bodnar, T. J., Brewer, D. D., Brownstein, J. S., Buckee, C., Campbell, E. M., Cattuto, C., Khandelwal, S., Mabry, P. L., et al. Digital epidemiology. *PLoS Comput Biol 8*, 7 (2012), e1002616.
6. Sebastiani, F. Machine learning in automated text categorization. *ACM computing surveys (CSUR) 34*, 1 (2002), 1–47.
7. Song, Y., Zhang, L., and Giles, C. L. Automatic tag recommendation algorithms for social recommender systems. *ACM Transactions on the Web (TWEB) 5*, 1 (2011), 4.
8. Wang, S., Lo, D., and Jiang, L. Inferring semantically related software terms and their taxonomy by leveraging collaborative tagging. In *Software Maintenance (ICSM), 2012 28th IEEE International Conference on*, IEEE (2012), 604–607.
9. Wang, S., Lo, D., Vasilescu, B., and Serebrenik, A. Entagrec: an enhanced tag recommendation system for software information sites. In *Software Maintenance and Evolution (ICSME), 2014 IEEE International Conference on*, IEEE (2014), 291–300.