Nearest Neighbour based Transformation Functions for Text Classification: A case study with StackOverflow

Piyush Arora, Debasis Ganguly and Gareth J.F.Jones
ADAPT Centre, School of Computing
Dublin City University, Dublin, Ireland
{parora,dganguly,gjones}@computing.dcu.ie

ABSTRACT

The significant growth in the number of questions in question answering forums has led to increasing interest in text categorization methods for classifying *newly* posted questions as *good* (suitable) or *bad* (otherwise) for the forum. Standard text categorization approaches, e.g. multinomial Naive Bayes, are likely to be unsuitable for this classification task because of: i) the lack of sufficient informative content in the questions due to their relatively short length; and ii) considerable vocabulary overlap between the classes. To increase the robustness of this classification task, we propose to use the *neighbourhood* of existing questions which are similar to the newly asked question. Instead of learning the classification boundary from the questions alone, we *transform* each question vector into a different one in the feature space. We explore two different neighbourhood functions using: the discrete term space, the continuous vector space of real numbers obtained from vector embeddings of documents. Experiments conducted on StackOverflow data show that our approach of using this neighborhood transformation can improve classification accuracy by up to about 8%. as compared to using just unigram textual features.

CCS Concepts

•**Information systems** → **Question answering;** *Query representation; Clustering and classification;*

Keywords

Neighbourhood based transformation; Document embedding; Question quality prediction

1. INTRODUCTION

Community question answering (CQA) platforms provide forums for knowledge exchange between individuals interested in a wide range of topics. The success of forums such as StackOverflow (SO) which currently contains around 11M questions in the area of software programming, make manual moderation of question quality a daunting task. CQA forums generally use a voting mechanism to obtain feedback from the community about the usefulness or suitability of each question. Based on this feedback, poorly scoring questions might be deleted or closed from the community. However, obtaining community feedback on questions takes time. Thus, it does not provide an indication of the quality of *newly* posted questions, which would for example enable them to be prioritised for attention or for deletion. A text categorization method that classifies such questions into: *good* (a question conforming to the community guidelines) or *bad* (a vague, imprecise, controversial question) would thus form a valuable component in the question moderation process. Despite the practical issue of the unavailability of community feedback for newly posted questions, to the best of our knowledge all prior work, e.g. [2], on CQA question classification has made use of community feedback, such as the number of votes, comments etc. Our work in this paper uses bag-of-words features from the questions, similar to [6] to address the problem of new question classification in CQA forums. Moreover, using only the text makes the approach generic enough to be applied to CQA forums which do not support community feedback, e.g. Ubuntu forums and Yahoo answers.

Carrying out this question classification task in the absence of community feedback faces two significant challenges: the relatively short length of the questions as compared to traditional web documents, and the considerable vocabulary overlap that exists between the *good* and *bad* questions. The complete SO question dataset, comprising $1.3M$ positive and $30K$ negative questions (see Table 1), has a high vocabulary overlap of 59.5% between the two classes. The sampled data for our experiments constituting 1000 samples from each class also has a high vocabulary overlap of 34.6%. In SO, the average length of questions in positive and negative classes is about 72 and 66 words, respectively. This high vocabulary overlap and relatively short length of questions indicates that a question by itself lacks sufficient informative and discriminative content for the classification task.

One obvious solution to this problem is to increase the number of training examples so as to incorporate more information into the classifier. However, this could lead to a strong imbalance in the number of samples between the positive and the negative class, because the number of down-voted questions in SO is much lower than the number of positively voted questions (see Table 1 and [6]). In this situation, adding more training examples is likely to bias the classifier towards predicting every question as *good*. The results of an exploratory investigation that we carried our on this imbalanced dataset (considering questions with at least 1000 views as in [6]) are shown in Table 2. These results indicate that increasing the number of training samples is not a good solution, because despite giving high accuracy, the average F-score (see Table 2) is close to that of random classification due to the strong class imbalance.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ICTIR '16, September 12–16, 2016, Newark, Delaware, USA.

© 2016 ACM. ISBN 978-1-4503-4497-5/16/09...$15.00

DOI: http://dx.doi.org/10.1145/2970398.2970426

Class name	Questions	
	All	#Views $\geq$ 1000
Bad (net score < 0)	380800	30163
Good (net score > 0)	3780301	1315731

Table 1: Frequency distribution for the two classes of SO questions (overall and the subset of questions with #views $\geq$ 1000). The net score of an SO question is the difference between the number of positive and negative votes.

Method	Accuracy	F-Measure (Macro average)
Only title content	0.9707	0.503
Title + Body content	0.9735	0.503

Table 2: Classification effectiveness for all SO questions with #views > 1000, #neg and #pos samples being 30,163 and 1,315,731, respectively (see Table 1) using MNB classifier.

To alleviate this problem of the lack of sufficient discriminative content in the questions, we propose to make use of other existing questions previously asked in the forum, to enrich the informative content of the question vectors. Since the previous questions asked on the forum already have votes, they can be treated as labelled samples and one may make use of a simple non-parametric classifier such as the K-NN. In the context of our problem, this simply refers to the process of retrieving other similar (in terms of text content) questions and classifying a question as *good* if it is surrounded by a higher number of positively voted questions than negatively voted ones. Unfortunately, such a simple classifier does not yield satisfactory results for our problem, as shown in Table 3. This motivates us to pursue the proposed method of transforming the data points before classification throughout the rest of the paper.

In contrast to using the class labels of other similar questions for K-NN classification, we propose to make selective use of other samples (whose labels we do not consider) for the purpose of transforming the current labelled sample in the training set. The objective is to obtain a different decision boundary than that obtained with the original training samples. Indeed, on each training instance vector (comprised of discrete terms or real numbers), $\mathbf{x}$, we propose to apply a *transformation* function ϕ to obtain a modified training sample $\phi(\mathbf{x})$. Recent research has shown that supervised learning on distributions rather than fixed training points proves to be effective, e.g, the support measure machine model proposed in [5]. Conceptually, the proposed transformation methods provide additional information from the neighbourhood which can be further used to form more descriptive features using kernel methods. The focus of this paper is to compare and explore the use of a transformation functions for the task of SO question classification. In the next section, we investigate the characteristics of this transformation function.

2. TRANSFORMATION FUNCTION

Our main hypothesis is that the transformation function ϕ operating on a vector $\mathbf{x}$ depends on the neighbourhood of $\mathbf{x}$. More specifically, ϕ is a function, say Φ, of the current point $\mathbf{x}$ and its neighborhood $N(\mathbf{x})$ as shown in Equation 1.

$$\phi(\mathbf{x}) = \Phi(\mathbf{x}, N(\mathbf{x})),\ N(\mathbf{x}) = \{\mathbf{x_i} : d(\mathbf{x}, \mathbf{x_i}) \leq r\} \qquad (1)$$

The transformation function shown in Equation 1 is a non-parametric one in the sense that this function solely depends on the current vector $\mathbf{x}$ and its neighbourhood, and cannot be expressed in a parametric form. In principle, the transformation function $\Phi(\mathbf{x}, N(\mathbf{x}))$ is somewhat similar to document expansion in information retrieval (IR), where a short document is expanded with the textual content from other documents in order to improve its informativeness and retrievability [3]. Further, from Equation 1, we can observe that the effectiveness of the transformation approach largely depends on the choice of the distance function $d(\mathbf{x}, \mathbf{x_i})$, the distance between a given training vector $\mathbf{x}$ and other vectors $\mathbf{x_i}$ that are not a part of the training set. An optimal choice of the distance function depends on the type of the vector $\mathbf{x}$, i.e. $\mathbf{x}$ is a real-valued or a categorical (e.g. composed of bag of words) vector. In particular for our study, we consider two different types of vector representations, namely the discrete term space representation of the SO questions (documents) (Section 4), and the finite dimensional vector representation learnt using neural network based embedding techniques such as [4] (Section 5).

k	Accuracy	F-score
1	0.4668	0.4766
3	0.4594	0.4548
5	0.4599	0.4523

Table 3: K-NN classifier fails to yield satisfactory results on a 50-50 train-test split of $1,000$ questions from each class.

3. EXPERIMENT SETUP

As introduced in Section 1, we used the 2014 StackOverflow data dump for our experiments. For all our classification experiments, in order to create a training set with reliable community feedback, we selected questions with at least a $1,000$ views, as prescribed in [6] and explored in our earlier work [1] . For the two classes in this classification task, we labeled questions with net positive scores (more up and then down votes) (*good*) and those with net negative scores (*bad*), as shown in Table 1. Due to the strong class frequency imbalance, using the whole dataset for training does not produce satisfactory classification effectiveness (Table 2 shows that despite high accuracy, the F-score values are close to random). Hence, we down-sampled the dataset by randomly selecting $1,000$ questions from each class to conduct our experiments.

In order to efficiently compute the neighbourhood (see Equation 1) for applying the transformation function of each question vector, each question in the SO dump, comprised of the title, body and the tags, was indexed with Lucene. For the neighbourhood set, we only consider questions that have non-zero scores. In total, the number of documents in the index (the questions with net zero scores were not indexed) is about 1.35M. For our text-based classification experiments, we use a Multinomial Naive Bayes (MNB) classifier with additive smoothing [7]. For document embedded vector based experiments, we use an SVM classifier (Gaussian kernel with default parameters $C = 1$ and $\gamma = 0.005$[1]). For all our experiments, we performed 10 fold stratified cross-validation over our dataset to avoid over-fitting. We evaluated classification performance using accuracy and F-score measures.

4. TERM BASED CLASSIFICATION

Neighbourhood selection using whole documents. We first performed standard text based classification with the help of the

[1]http://scikit-learn.org/stable/

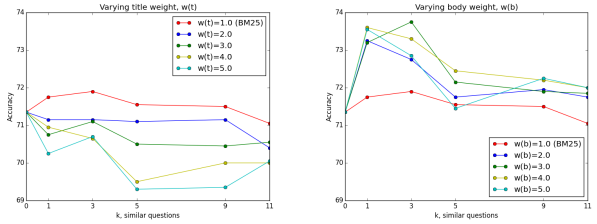

Figure 1: Accuracies obtained with different neighbourhood transformations and weights for the title (left, $w(B) = 1$) and the body (right, $w(T) = 1$) fields of SO questions.

Table 4: Multinomial Naive Bayes classification with BM25 based neighbourhood transformation.

k	Accuracy	F-Measure
0	0.713	0.704
1	0.718	0.710
3	**0.719**	**0.713**
5	0.715	0.710
9	0.715	0.711
11	0.710	0.706

MNB classifier by making use of text of each question in the training set. This amounts to simply using the original training vectors $\mathbf{x}$ for learning a decision boundary (reported as the result $k = 0$ in Table 4). Next in order to test our hypothesis that enhancing the informative content of a question improves classification effectiveness (see Section 2), we undertook a simple approach of finding $N(\mathbf{x})$, the neighbourhood of a data point $\mathbf{x}$. We considered the title of a question as a query and retrieved a ranked list of related questions from our SO index. For retrieval we used BM25, with parameters $k = 1.2$ and $b = 0.75$. The BM25 similarity measure acts as the inverse distance metric d of Equation 1. The top k documents from this list yields the desired neighbourhood. As the transformation function $\Phi(\mathbf{x}, N(\mathbf{x}))$, we concatenate (denoted by the $\oplus$ operator) the text of each $\mathbf{x_i} \in N(\mathbf{x})$ to $\mathbf{x}$.

$$\Phi(\mathbf{x}, N(\mathbf{x})) = \mathbf{x} \oplus (\oplus_{x_i \in N(\mathbf{x})} \mathbf{x_i}) \quad (2)$$

We also varied the parameter k, i.e. the number of top most similar documents (SO questions), used to define the neighbourhood for transforming the current document (SO question) before MNB classifier training. $k = 0$ refers to the 'no transformation' case. The results in Table 4 show improvement in classification effectiveness (compare the F-scores with the baseline case, i.e. when $k = 0$ and those of Table 2). This empirically proves the hypothesis that it is possible to achieve a better decision boundary between the two classes of SO questions by transforming each training vector $\mathbf{x}$ to a different point $\Phi(\mathbf{x}, N(\mathbf{x}))$ similar to the results shown in our earlier work [1]. Another interesting observation is that with $k = 11$, classification effectiveness decreases. This happens because too large a neighbourhood is prone to attract noisy information for classifier training.

Neighbourhood selection using individual document fields. After observing improvements in classification results with the use of neighbourhood, we now explore alternative ways of obtaining potentially better neighbourhoods and transformation functions by leveraging the content of the individual fields of a question, namely its title and body. Instead of using BM25 (as used for selecting neighbourhoods based on the whole question text), for the field based neighbourhood selection, we applied BM25F [8] which makes provision to associate relative importance to each field by assigning individual weights to each. To search for the optimal neighbourhood for a question, a query was formulated by using the terms from the 'title' field and retrieval was performed over documents comprised of two fields - the 'title' and the 'body'. To find the optimal settings for the relative weights, denoted $w(B)$ and $w(T)$ corresponding to the body and the title respectively, we performed a two-step grid search as shown in Figure 1. We fixed one of the weights and varied the other in the range $[1, 5]$. The best results were obtained with $(w(T), w(B)) = (1, 3)$ with $k = 3$.

Table 5: Multinomial Naive Bayes classification with BM25F based neighbourhood transformation.

k	Neighborhood	Accuracy	F-Measure
0	N/A	0.713	0.704
3	BM25	0.719	0.713
3	BM25F	**0.738**	**0.733**

Setting a higher weight for w_T (the weight of the title field) degrades classification accuracy. The most likely reason is that the top ranked retrieved questions defining the neighbourhood of $\mathbf{x}$ may only have a few matching terms with the current document from the title field only. A few matching terms in the title may not necessarily mean that the current question and those in the neighbourhood are topically related to each other. On the other hand, the main informative content of a SO question is likely to be present in the body field. Matching the title terms of the query (a new question) with those in the body of previous questions proves to be more effective in retrieving topically related questions.

Table 5 compares the results obtained with optimal BM25F parameter settings (as per Figure 1) with the BM25 based neighbourhood (reproduced from Table 4). We find that the accuracy obtained with BM25 based neighborhood transformation increases by 2.6% and the F-score by 2.8% in comparison to the BM25 one. This not only shows that BM25F is a better (inverse) distance measure for choosing the neighbourhood function $N(\mathbf{x})$ more effectively, but also indicates that the neighbourhood function plays a crucial part in estimating the decision boundary for this classification problem.

5. EMBEDDING BASED CLASSIFICATION

In this section, we approach the SO question classification using vector embeddings of documents [4]. Embedded representations have shown to outperform text based approaches for various NLP tasks, such as sentiment analysis [4]. The document vector embedding process learns a one-hot vector representation of a word or sentence. These vectors can then be used as inputs to a classifier.

5.1 Embedding SO questions as vectors

In the context of our classification task, the purpose of document embeddings is to represent the semantic relatedness between documents with the help of distances between real valued vector. An interesting question for our study is then: how may we choose the neighbourhood and the transformation functions for real valued vectors rather than text? In particular, we used the model of distributed representation of sentences and documents [4] to learn embeddings for questions. We used gensim implementation[2] of *doc2vec* for training the model and learning question embeddings.

[2]https://radimrehurek.com/gensim/models/doc2vec.html

For each of the question q in our data set, we learned a real-valued vector with the component values ranging between $[-1, 1]$. For real-valued vectors, one cannot use Multinomial Naive Bayes (MNB) for classification as it uses MLE estimates of categorical features, such as terms of a document. For classification experiments with the document vector representations, we used the SVM classifier. As a baseline approach we applied the SVM classifier on the vector embeddings of documents to obtain the decision boundary. To obtain the document embedded vectors, we experimented with both the 'distributed bag of words' ($dbow$) model and the 'distributed memory model' (dmm) as described in [4]. The SVM classification effectiveness obtained with the $dbow$ document vectors outperforms those obtained with the dmm model, as a result of which, all the subsequent results reported in this section used the $dbow$ model trained document vectors for classifier training. Another parameter for the document vector embedding process is the dimensionality of the vectors (i.e. the number of nodes in the output layer of the neural network). We varied this number from 100 to 500 with a step size of 100. We obtained the best classification results with 200 dimensional embeddings. The 'window size' parameter for the document embedding process was set to 10 as prescribed in [4].

5.2 Neighbourhood and Transformation

For document embedded vectors, we applied the most common notion of similarity, the inner product between two vectors, as the (inverse) distance metric for choosing the neighbourhood around a vector $\mathbf{x}$ (see Equation 1). Note that for real-valued vectors we cannot make use of BM25 similarity as used in the experiments of Section 4. After obtaining the neighbourhood set $N(\mathbf{x})$, the next step is to apply the Φ function, which defines how to use the information contained in the neighbourhood set to enrich the current data point. A standard way to combine a set of vectors in $\mathbb{R}^P$ is to compute the centroid vector. We considered the more generalized approach for computing the weighted centroid, where each point $\mathbf{x_i}$ from the neighbourhood set is weighted by its distance from the current point $\mathbf{x}$, as shown in Equation 3.

$$\Phi(\mathbf{x}, N(\mathbf{x})) = \mathbf{x} + \sum_{\mathbf{x_i} \in N(\mathbf{x_i})} (\mathbf{x} \cdot \mathbf{x_i})\mathbf{x_i} \quad (3)$$

It can be seen that the dot product between $\mathbf{x}$ and its neighbour $\mathbf{x_i}$, which is a real number, acts as the weight to define the relative contribution of $\mathbf{x_i}$ in the transformation function. Unlike the transformation function for the text space (Equation 2), the weighted neighbourhood for document embedded vectors (Equation 3) is able to capture the contributions from each individual point of the neighbourhood in variable quantities.

Table 6 shows the results after applying the transformation function Φ, as defined in Equation 3, on the embedded document vectors after training with SVM. The baseline case, i.e., when no transformation is applied, is shown with $k = 0$. Similar to the text based classification, we varied the neighbourhood size to see its effect on classification performance. From Table 6, we observe that the application of the transformation function shows consistent improvement for the document embedded vectors as was the case for the text vectors (see Table 5). The best classification effectiveness achieved with the vector embeddings of SO questions is higher than that obtained with the text features (compare Table 6 with Table 5).

Table 6: SVM classification after applying neighborhood based transformation on document vector embeddings.

k	Accuracy	F-Measure
0	0.743	0.743
1	0.740	0.739
3	0.747	0.746
5	0.750	0.749
9	**0.769**	**0.768**
11	0.765	0.764

6. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a general framework for applying a non-parametric based transformation function on a given set of training examples before training a parametric classifier. The transformation function decides what information to use from the local neighbourhood of a training point so as to enrich the representation of the current point. The task that we address is the classification of a newly posted StackOverflow question as good or bad for its community. The considerable vocabulary overlap between the classes and short length of questions make this a challenging classification task, which is an ideal candidate for the transformation function based approach to classification. We study two different classification approaches, one with text and the other with document vector embeddings. We find empirically that selecting a good neighbourhood is important for achieving good classification results. Results show that the use of BM25F with more importance given to the match in the 'body' field of a question results in a better neighbourhood estimation with improved classification. For the text based approach, we applied concatenation as a transformation function. The neighbourhood is computed using BM25 similarity. For the document vector space, the neighbourhood is computed using dot products coupled with a weighted centroid transformation function. Consistent trends in improvements of classification results are observed with the transformation function applied on both text and document vector embeddings. The improvements in classification F-scores obtained with the transformation function on the text and on the document embedded vectors are up to 4.1% and 3.4%, respectively. As a part of future work, we would like to explore alternative transformation functions, and different ways of combining the neighbourhood and the transformation functions of the textual and the document vector spaces.

Acknowledgment: This research is supported by Science Foundation Ireland (SFI) as a part of the ADAPT Centre at Dublin City University (Grant No: 12/CE/I2267 and 13/RC/2106).

7. REFERENCES

[1] P. Arora, D. Ganguly, and G. J. Jones. The good, the bad and their kins: Identifying questions with negative scores in stackoverflow. In *Proceedings of ASONAM '15*, pages 1232–1239. ACM, 2015.

[2] D. Correa and A. Sureka. Chaff from the wheat: characterization and modeling of deleted questions on stack overflow. In *Proceedings of WWW '14*, pages 631–642, 2014.

[3] M. Efron, P. Organisciak, and K. Fenlon. Improving retrieval of short texts through document expansion. In *Proceedings of the SIGIR '12*, pages 911–920, 2012.

[4] Q. V. Le and T. Mikolov. Distributed representations of sentences and documents. In *Proceedings of ICML '14*, pages 1188–1196, 2014.

[5] K. Muandet, K. Fukumizu, F. Dinuzzo, and B. Schölkopf. Learning from distributions via support measure machines. In *Proc. of NIPS '12*, pages 10–18, 2012.

[6] S. Ravi, B. Pang, V. Rastogi, and R. Kumar. Great Question! Question Quality in Community Q&A. In *Proc. of ICWSM '14*, 2014.

[7] J. D. Rennie, L. Shih, J. Teevan, and D. R. Karger. Tackling the poor assumptions of naive bayes text classifiers. In *Proc. of ICML '03*, pages 616–623, 2003.

[8] S. E. Robertson, H. Zaragoza, and M. J. Taylor. Simple BM25 extension to multiple weighted fields. In *Proceedings of CIKM '04*, pages 42–49, 2004.