Duplicate Question Detection using Online Learning

Chapman Siu

College of Computing, Georgia Institute of Technology

Atlanta, GA 30332-0280, USA

Email: chapmansiu@gatech.edu

Abstract

An integral part of OMSCS learning experience are online communities such as Piazza. Piazza is described as a "Q&A" forum which is created for every class[1]. An integral part of many online communities is the concept of "voting"' to determine popular and useful information. However, Piazza lacks this particular functionality which is on other Q&A boards like Stackoverflow or Quora.

Being able to understand quality posts can be beneficial for students and instructors, as it will reduce the clutter in the community by allow students to "see"' important posts in the follow up section of different threads and for instructors to identify potential experts and teaching assistant candidates for future classes. By building a corpus of high quality questions, this could even be extended to develop answers to user questions.

In this project, we aim to develop suitable approach for understanding quality of posts through introducing online machine learning techniques to assist with reducing clutter within young online communities.

Keywords

duplicate question, Piazza

I. Introduction

To understand and be able to explain quality posts, the type of modelling we are after is explanatory modelling, where we seek to provide causal explanations. One key consideration is that Shmueli[3] describes how explanatory modelling does not always generalise well to

predictive models. Shumueli[3] also notes that the choice between explanatory and predictive modelling may affect the type of variables you use (variables have to make sense from a causal perspective), the type of preprocessing you may perform (data compression methods such as SVD may be inappropriate from a causal perspective), types of models you may build (ensemble of models would be inappropriate from a causal perspective).

Comparing with previous studies, we are aiming to demonstrate how these techniques can be used in an online learning context. Online learning focuses on modelling data which arrives in sequential order to update our best predictor for future data as opposed to training on the whole training data set at once. This is important part of this project as with young communities, often the whole training data set will not be available until the community is relatively mature. This research would be similar to Zhang *et al*[6] conducted similar experiment with duplicate question detection via batch learning. Pal *et al.*[2] explains that early identification and engagement with these users can improve the experience of these users and more importantly improve the overall quality of participation within a community. Within Pal *et al.*[2] retrospective analysis, it had already shown that some of the potential experts had already left the community when the analysis concluded leading to missed opportunities within the community.

Furthermore to understand the type of questions which a community is asking can allow the noise to be diminished. For example on roughly 13 October 2016 the OMSCS admission results for Spring 2017 intake was annouced. Out of the newest 15 posts, 9 of them were new students questions or notes, with two threads having over 60 posts altogether. Similarly, reddit had 5 of 7 posts within the last 24 hours about new admissions, with 42 out of 44 comments in the respective posts in the threads on admissions.

Having adaptive duplication detection not only assists in special events but also in young communities, where there is a lack of training data or examples to create a sensible detection for duplicated questions. On forums like Piazza this would assist in reducing the amount of clutter which exists, similar to how stackoverflow closes duplicate questions for redundant questions to decrease the maintenance and people's resources on answering the same question[6].

The following usage scenarios demonstrate the benefits of such a tool.

Scenario 1 - *Without tool.* Steve recently gained admission to OMSCS program. However he is confused by the registration process and is unsure which course he should pick. He posts to the Google Plus community but received no response due to the flood of registration related posts.

Scenario 2 - *With tool.* Steve recently gained admission to OMSCS program. However he is confused by the registration process and is unsure which course he should pick. He posts to the Google+ community. By using our automated tool, we can readily detect similar questions that have been asked before and direct Steve to a useful resource.

We evaluate our approach on synthetically generated posts based on real questions asked on Google+ and Reddit, combined with unofficial FAQs generated by the community. We will also evaluate the performance by simulating a newly created Piazza forum to see if the tool has a similar efficacy.

The main contributions of this paper are as follows.

- We propose the problem of duplicate question detection in online communities. We procose a novel approach which considers and integrates multiple factors to detect duplicate questions.
- We evalute different types of communities based on real and synthetic questions.

The remainder of the paper is organised as follows. We elaborate on the motivation of our work and introduce online learning variants of Latent Dirichlet Allocation (LDA) and Word2Vec, and describe the other components of the overall framework. Next we will discuss some issues about the performance, efficiency and threats to validity. We will review related work, conclude the paper and mention future work.

II. PRELIMINARIES

A. Online Latent Dirichlet Allocation

Online Latent Dirichlet Allocation is an online variant of Latent Dirichlet Allocation (LDA). LDA is a well-known topic modelling technique proposed by Blei *et al.*[7]. LDA is a generative probabilistic model of a textual corpus (i.e., a set of textual documents), which takes a training textual corpus as input, and a number of parameters including the number of topics (K) considered. In the training phase, for each document s, LDA will compute its topic distribution θ_s, which is a vector with K elements, and each element corresponds to a topic. The value of each element in θ_s is a real number from 0 to 1, which represents the proportion of the words in s that belong to the corresponding topic in s. After training, LDA can be used to predict the topic distribution θ_m of a new document m. In our case, a document is the description of a question, and the topic is a higher level concept corresponding to a distribution of words. For example, we may have the

topic ”admissions”, which is a distribution of words such as “citizenship”, “GRE”, “TOEFL”, “transcripts”.

LDA can be extended in an online learning problem by reframing LDA using approximate inference techniques (variational inference) which then becomes an optimization problem and allows LDA to be trained via known approaches such as stochastic gradient descent[8].

B. Online Word2Vec

Word2Vec is all about computing distributed vector representations of words. In this project we will be using the skip-gram variant.

The training objective of skip-gram is to learn word vector representations that are good at predicting its context in the same sentence. Mathematically, given a sequence of training words $w_1, w_2, ..., w_T$, the objective of the skip-gram model is to maximize the average log-likelihood

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{j=-k}^{j=k}\log \Pr(w_{t+j}|w_t)$$

where k is the size of the training window.

In the skip-gram model, every word w is associated with two vectors u_w and v_w which are vector representations of w as word and context respectively. The probability of correctly predicting word w_i given word w_j is determined by the softmax model, which is

$$\Pr(w_i|w_j) = \frac{\exp(u_{w_i}^T v_{w_j})}{\sum_{l=1}^{V}\exp(u_l^T v_{w_j}}$$

where V is the vocabulary size.

The skip-gram model with softmax is expensive because the cost of computing $\log(\Pr(w_i|w_j))$ is proportional to V, which can be easily in order of millions.

Online variant of Word2Vec solves the online learning problem, by updating vocabulary whenever new documents are ingested. The new words are then initialized with random weights, whilst existing words retain their weights as normal. Then training will be again be initialised in an iterative fashion.

C. Cosine Similarity

Cosine similarity is used to compute the similiarty between pairs of sets of words based on common words that they share. After preprocessing, the words are transformed into two bags (i.e. multisets) of words. For two sets of words m and n, we represent the two bags of words that extracted as Bag_m and Bag_n respectively. Next we merge Bag_m and Bag_n and eliminate duplicate words to obtain the union set Bag_u, which contains v words. Following vector space modelling, we represent the two sentences as two vectors: $Vec_m = (wt_{m,1}, wt_{m,2}, ..., wt_{m,v})$ and $Vec_n = (wt_{n,1}, wt_{n,2}, ..., wt_{n,v})$. The weight $wt_{q,i}$ denotes the relative term frequency of the i-th word in sentence q's title, which is computed as follows:

$$wt_{q,i} = \frac{n_{q,i}}{\sum_v n_{q,v}}$$

where $n_{q,i}$ denotes the number of times the i-th word of Bag_u appears in the sentence q, $\sum_v n_{q,v}$ denotes the total number of occurences of all words in the title of question q, where v is the index of the word in Bag_u. We measure the similarity between two questions' titles by computing the cosine similarity of their vector representations Vec_m and Vec_n as follows:

$$CosineSim(Vec_m, Vec_n) = \frac{\langle Vec_m, Vec_n \rangle}{|Vec_m||Vec_n|}$$

The numerator $\langle Vec_m, Vec_n \rangle$ which is the dot product of the two vectors

$$\langle Vec_m, Vec_n \rangle = wt_{m,1} \times wt_{n,1} + ... + wt_{m,v} wt_{n,v}$$

The terms $|Vec_m|$ and $|Vec_n|$ in the denomiator denote the sizes of the two vectors respectively, where the size of Vec_m is computed as

$$|Vec_m| = \sqrt{wt_{m,1}^2 + wt_{m,2}^2 + ... + wt_{m,v}^2}$$

Cosine similarity measures do not require the whole corpus in order to compute similarity; rather it depends only on the pairwise sets of words. As such there is no online variant to cosine similarity as one does not need to build or store a corpus in its formulation.

III. Proposed Approach

A. Overall Framework

Insert some image of the framework and description of how it would work overtime.

B. Feature Construction

Probably revolves around the more qualitative side, as described here: http://researcher.ibm.com/researcher/files/us-apal/umap11_earlyexperts.pdf

C. Estimation of Weights

Logistic regression would probably be good enough...

IV. Experiments and Results

A. Experimental Setup

Our online learning model is evaluated on historical questions in Stack Overflow. We parse data for 'html' and 'javascript' tags from January 2008 to October 2016 using Stackexchange API. We extract approximately 750 000 questions over 10 month period.

B. Evaluation Metrics

To evaluate the performace, we will use recall-rate which is also used in [6]

$$\text{recall-rate} = \frac{N_{\text{detected}}}{N_{\text{total}}}$$

C. Research Questions and Findings

How effective? How does it compare to a batch learning approach.

V. Discussion

A. Threats to Validity

Threats include the ability of this approach to generalise to all young communities. As young communities is quite broad, the focus in this paper has been quite targeted, looking at computer science forums, and learning centers. The duplicate questions were also quite targeted and manually pruned. In an more unbalanced set, this ....?

Threats to construct validity - suitability of evaluation metrics

VI. Related Work

Besides what has been in the proposal, links to online learning, such as vowpal wabbit:

VII. Conclusion

Some conclusion

Appendix A
Proof of the First Zonklar Equation

Some text for the appendix.

References

[1] Kincaid, J. (2010, February 5). "Piazzza gives classmates an online forum to trade their knowledge"'. Retrieved September 18, 2016, from https://techcrunch.com/2010/02/05/piazzza-college-questions-answers/

[2] Pal, A., Farzan, R., Konstan, J. A., and Kraut, R. E. "Early detection of potential experts in question answering communities." in *User Modeling, Adaption and Personalization*. Springer, 2011, 231-242. Retrieved on 12 September 2016 from http://researcher.ibm.com/researcher/files/us-apal/umap11_earlyexperts.pdf

[3] Shmueli, G. (2010). "To explain or to predict?" *Statistical Science* 25(3), 289310. doi:10.1214/10-sts330. Retrieved on 30 September from http://www.stat.berkeley.edu/~aldous/157/Papers/shmueli.pdf

[4] Manning C.D., Raghavan P. and Schtze H. (2008), "Introduction to Information Retrieval", Cambridge University Press. 2008. Retreived on 30 September from http://www-nlp.stanford.edu/IR-book/

[5] Bogatyy I. (2016) "Predicting answer types for question-answering". Retrieved on 30 September from https://cs224d.stanford.edu/reports/Bogatyy.pdf

[6] Zhang, Y., Lo, D., Xia, X., and Sun, J.-L. (2015). "Multi-factor duplicate question detection in stack overflow'. *Journal of Computer Science and Technology*, 30(5), 981997. Retrieved on 17 September 2016 from https://soarsmu.github.io/papers/jcst-duplicateqns.pdf. doi:10.1007/s11390-015-1576-4

[7] Blei, D. M., Ng, A. Y. and Jordan, M. I. (2003). "Latent dirichlet allocation". J. Mach. Learn. Res., 3, 993–1022. doi:10.1162/jmlr.2003.3.4-5.993. Retreived on 12 October from www.jmlr.org/papers/volume3/blei03a/blei03a.pdf

[8] Hoffman, M. D., Blei, D. M. and Bach F. (2010). "Online Learning for Latent Dirichlet Allocation" Retreived on 12 October from http://www.cs.princeton.edu/~blei/papers/HoffmanBleiBach2010b.pdf

Chapman Siu is Masters candidate from Georgia Institute of Technology. Chapman is also a lion.