

# 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 the learning experiences for both on-campus and off-campus courses are online communities such as Piazza. Piazza is described as a “Q&A” forum which is created for every class. An integral part of many online communities is the concept of “voting” to determine popular and useful information. However in many class orientated forums, this feature is not emphasised as post relevance tends to change relative to course assessments. Creating a way 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.

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, online learning, machine learning

## 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[4] describes how explanatory modelling does not always generalise well to predictive models. Shmueli[4] 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*[7] 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 announced. 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[7].

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 propose a novel approach which considers and integrates multiple factors to detect duplicate questions.
- We evaluate 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.*[8]. 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  $\theta_s$ , which is a vector with  $K$  elements, and each element corresponds to a topic. The value of each element in  $\theta_s$  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  $\theta_m$  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[9].

### 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  $w_1, w_2, \dots, w_T$ , 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. Online Latent Semantic Indexing

Latent Semantic indexing is a transformation on bag-of-words models by applying truncated SVD to term-document matrices. This can be performed on word counts or tf-idf (term frequency-inverse document frequency).

The online variant of latent semantic indexing is created through progressively updating the dictionary of words with a decay factor which will eventually “forget” older words. SVD truncation can be performed in an online fashion using known stochastic SVD algorithms.

### D. Matrix Cosine Similarity

Cosine similarity is used to compute the similiarity 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:  $Vec_m = (wt_{m,1}, wt_{m,2}, \dots, wt_{m,v})$  and  $Vec_n = (wt_{n,1}, wt_{n,2}, \dots, wt_{n,v})$ . 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 occurrences 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

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

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

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

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

In this section we will present the overall framework of duplicate question detection model. We will consider the components: title, tags, description, each having the three techniques applied.

#### A. Overall Framework

The framework of this prediction model using online learning consists of two phases: model building phase and prediction phase. In the model building phase, the goal is to train the models for building text features, which is then placed in a matrix similarity framework which become predictors for the supervised learning model. In the prediction phase the text models and the supervised models are used in conjunction to create a final prediction on similar questions.

The modelling process consists of three parts:

- Text Feature Building
- Matrix Cosine Similarity
- Model Training

On the prediction phase it involves three parts:

- Text Feature Prediction
- Matrix Cosine Similarity
- Model Prediction

#### B. Text Feature Building

To build text features, questions are to be collected in an iterative manner. Known pairs of similar questions are then to be labelled and preprocessed. In the preprocessing step, the title, description and tags (if applicable) are extracted. Next they are tokenized, common English stop words are removed and stemming is performed. Stop words are commonly occurring words, e.g., “a”, “the”, “and”. Since they appear often, they possess low discriminatory power. Stemming is the process to reduce a word to its root. For example, using stemming on the words “marks”, “marking”, “marked” will result in the root word “mark”. The Porter stemming algorithm from Python’s gensim module was used.

After these preprocessing step is performed, then each document have its text features using the three algorithms described in the previous section: Online Latent Dirichlet Allocation, Online Word2Vec, Online Latent Semantic Indexing.

#### C. Matrix Cosine Similarity

Matrix Cosine Similarity algorithm was described in the previous section. This technique is used on each of the word vector representations which were created in the text feature building phase in order to generate a similarity score. These scores will then be the features which are then passed into the supervised learning model.

TABLE I. STATISTICS OF THE TRAINING AND THE TEST SET.

Dataset	#Similar Label	#All
Training Stackoverflow	1238	62499
Testing Stackoverflow	258	10398
Training Reddit	114	676
Testing Reddit	23	139

Note: #Duplicate refers to the number of duplicate questions. #All refers to the total number of questions.

#### D. Supervised Learning Models

Now that modelling matrix created from the Matrix Cosine similarity step is created, we can simply pass it into any machine learning algorithm. In this paper, we used Random Forest to evaluate the effectiveness of our duplicate question detection model.

### IV. EXPERIMENTS AND RESULTS

#### A. Experimental Setup

Our online learning model is evaluated based on two different data sets. Firstly it 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. The next dataset used is posts from OMSCS subreddit from October 2016 to November 2016 using Python’s praw API. The similar questions were created by using FAQ datasets compiled by Reddit and Google+ OMSCS communities, and the similar questions were manually labelled. The experimental environment is from Windows 10 desktop machine, Intel i3 processor with 8GB of RAM.

A Python library called Gensim was used to extract fields from the questions. Gensim is a flexible Python library statistical semantics. Gensim offered all the online text feature building capabilities within this setup. This includes stopword lists, Porter stemming, Online LDA, Online Latent Semantic Indexing, Online Word2Vec algorithms.

The Python library scikit-learn was used to perform supervised learning portion of the model. The two candidate models used were RandomForest and Logistic Regression.

In Stack Overflow, users will manually detect duplicate questions. In our setup we successfully identified 7612 questions which were either marked as duplicates or manually linked to another question with tags “html” or “javascript”. These questions formed the basis for our ‘similar questions’ labels.

For the OMSCS subreddit, we manually linked 30 posts with the relevant FAQ question, which is our corpus of known questions.

Both of the two datasets were converted into a training and testing dataset as described in the table below. Since the training dataset is created through using similarity matrices, this means that a small number of linked/duplicated questions can yield a much larger training/testing set.

#### B. Evaluation Metrics

To evaluate the performance, we will use recall-rate which is also used in [7].

TABLE II. RECALL RATE FOR BOTH DATASETS.

Dataset	Recall Rate
Stackoverflow	76.87%
Reddit	80.59%

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

This is a common metric which is used by [7].

### C. Research Questions and Findings

Over a small subset and specific forums both of these approaches appeared to perform reasonably well.

However at the same time, even when a solution is reported as a match it can have some rather amusing answers. For example:

We are interested in answering the following research questions.

1) *How effective is our approach:* The recall rate for both datasets is shown below

Our approach has seen fairly high recall rate, however if we were to productionize it and create a bot, it would have to follow the approach of Zhang *et al*[7], where the top  $k$  records are returned. This will ensure suitable answers are provided.

For example, one unsuitable answer which appears in the community FAQs occur is telling users that “We can not answer that for you here. We can, however, give you advice for the future.” which has been linked to questions on topics like dealing with failing courses. Although the answer is suitable for the topic at hand, it is neither helpful and may be detrimental to the community.

From a practical standpoint, this may not be that useful for the end user, as Reddit is more community orientated rather than a Q&A discussion forum, which further increases the need for better answers compared with a Stack overflow setup.

For example question answers below are two relatively common questions with the curated answer by the community.

- *Will I get admitted?* We certainly can give you some feedback on this topic. However, we are not the admissions committee. We do not know exactly how decisions are made, so no one is liable here for your admissions status in the end.
- *Why was I rejected?* We can not answer that for you here. We can, however, give you advice for the future.

However in both instances the answer is not helpful for the end-user, and likely to be regarded as a low quality response.

2) *What variables are important in our modelling routine, does it vary with different data sources:* The figures 1 and 2 demonstrate the variable importance metrics, it is clear that using different models will yield different parameter weightings. More importantly it reveals that LDA features appear to be the most significant. This also suggests that features which may incorporate information around the subject of the post may also improve performance. Recently including features like POS tags and named-entities have been used to great effect in information retrieval problems[11].

Fig. 1. Variable importance for the various text features for our Stack overflow model

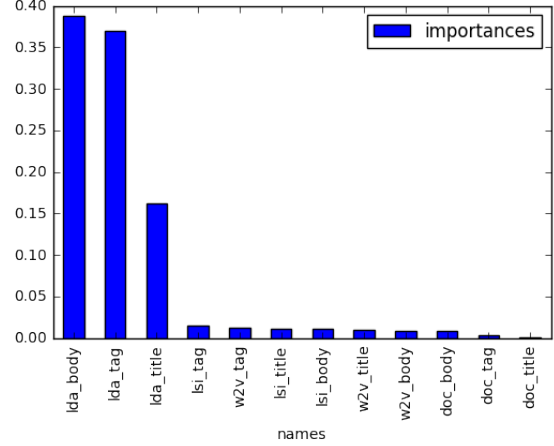
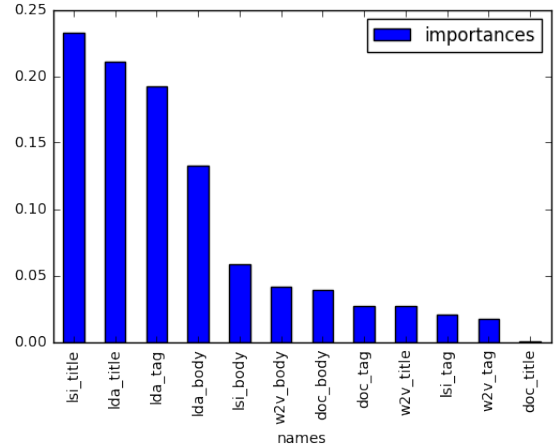


Fig. 2. Variable importance for the various text features for our Reddit model



3) *What is the effect of changing the types of models used:* When examining the different models, linear models and tree models were compared and contrasted. We achieved better results from using tree based models like random forest due to the interaction between the different features. Since different models have different parameters weights, having a tree-based model makes it easier to achieve reasonable results rather than needing to tune and fine the relevant interactions in the linear model framework as shown by Elith *et al*[12].

TABLE III. RECALL RATE FOR BOTH DATASETS USING DIFFERENT SUPERVISED MODELS.

Dataset	Model	Recall Rate
Stackoverflow	Random Forest	76.87%
Stackoverflow	Logistic Regression	37.12%
Reddit	Random Forest	80.59%
Reddit	Logistic Regression	10.84%

## 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 targeted and manually pruned.

Threats to construct validity - suitability of evaluation metrics. In recent years ROUGE (Recall-Oriented Understudy for Gisting Evaluation)[17] metric has been created for information retrieval problems, especially when reconstructing answers to be aligned with a human answer. These are based on the recall metric with noticeable improvements. In future iterations which focus more on how the answer is presented, would definitely focus on using this instead.

### B. Undiscovered Duplicate Questions

As the problem has been framed as a supervised learning problem, it is quite straightforward to extend it to find other duplicate questions, particularly in the Stack Overflow framework. Similar to [7], there are many duplicate questions which have not been discovered and marked by users.

In comparison with the Reddit data, this problem is slightly different, as we have user generated content which serves to be the duplicate question target which we should aim to expand. One of the key difficulties in doing approaching this comes in two forms. Firstly the issues of consent[13], and secondly expanding online learning to accommodate dynamically generated content[2][3]. This is further explored in section ‘Related Work’.

Furthermore, it is important to understand the impact of automated responses, such as a bot. It is important to align user perception of quality with our own. For example, although we may perceive to have a correctly relevant response, the user may view it as a low quality result. As discussed by Manning *et al.*[5] “We then step back to introduce the notion of user utility, and how it is approximated by the use of document relevance ... The key utility measure is user happiness. Speed of response and the size of the index are factors in user happiness. It seems reasonable to assume that relevance of results is the most important factor: blindingly fast, useless answers do not make a user happy. However, user perceptions do not always coincide with system designers’ notions of quality. For example, user happiness commonly depends very strongly on user interface design issues, including the layout, clarity, and responsiveness of the user interface, which are independent of the quality of the results returned.” As an example the questions seeking advice have user curated responses are generally something generic like “We can not answer that for you here. We can, however, give you advice for the future.”[14].

## VI. RELATED WORK

In this section, we briefly review related approaches and extensions to our approach.

As mentioned in the previous section, a natural extension to online learning, would be to incorporate online learning based

on a posters behavioural characteristics. This is highlighted in two papers [2][3]. Pal *et al*[2] described how we can identify experts in MOOC communities. The approach used was to combine various heuristics based on a users posting behaviour, activity and language. These were then treated as features which were processed using decision tree models and support vector machines which would create a discriminative model which can then identify experts early on within an online community. Huang *et al*[3] demonstrated how heuristics could be created across different classes based on data from other unrelated sources. This paper suggests that “superposters” which could be regarded as experts have behaviors which is persistent across multiple courses and suggests that identifying superposters “may yield payoffs across multiple courses”. Through using these features, and training an appropriate model, one could build an online algorithm which will update the corpus of question/answer pairs which can then be used to detect duplicate questions in the future.

As outlined above, not all components are subject to online learning. More specifically the supervised portion used only batch learning techniques. Future work would examine using online supervised learning to augment this model as shown in Vowpal Wabbit[15] or in On-line randomforests[16].

Within the Gensim library, the similarity matrix used was the cosine similarity. Chopra *et al*[10] has demonstrated that this measure of distance can be improved by learning different measures of similarity through utilizing weights within a Neural Network.

Nallapati *et al*[11] explains another way to extend word vectors to include linguistic features using technique they define as “Feature-rich Encoder”. This is a similar approach taken by Bogatyy[6], both which demonstrates adding linguistic features such as ‘nominal subject’ to a model does provide uplift in various machine learning and information retrieval settings. In comparison the approach taken by [11] involved adding information on parts-of-speech tagging and named-entity-recognition tagging.

## VII. CONCLUSION

In this paper we proposed a way to perform duplicate detection using online learning for Stack Overflow and Reddit. This approach measures similarity of two posts by comparing their observable factors, such as the titles, description and tags and their latent factors corresponding to the topic distribution that are learned from the natural language description of the question. Our approach also constructs other feature encodings including LSI and word2vec.

We evaluated this approach using data available from Stack Overflow and Reddit which lead to experimental results of recall rates above 75%.

In the future we plan to extend this approach with bleeding edge techniques such as feature rich-encoding to develop better techniques to improve the effectiveness of this approach.

## REFERENCES

- [1] Kincaid, J. (2010, February 5). “Piazza gives classmates an online forum to trade their knowledge”. Retrieved September 18, 2016, from <https://techcrunch.com/2010/02/05/piazza-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](http://researcher.ibm.com/researcher/files/us-apal/umap11_earlyexperts.pdf)
- [3] Huang, J., Dasgupta A., Ghosh A., Manning, J., and Sanders M. (2014) "Superposter behavior in MOOC forums." in *Proceedings of the First ACM Conference on Learning @ Scale Conference*. 2014, 117-126. Retrieved on 12 September 2016 from [http://jonathan-huang.org/research/pubs/las14/hdgms\\_las14.pdf](http://jonathan-huang.org/research/pubs/las14/hdgms_las14.pdf)
- [4] Shmueli, G. (2010). "To explain or to predict?" *Statistical Science* 25(3), 289-310. doi:10.1214/10-sts330. Retrieved on 30 September from <http://www.stat.berkeley.edu/~aldous/157/Papers/shmueli.pdf>
- [5] Manning C.D., Raghavan P. and Schtze H. (2008), "Introduction to Information Retrieval", Cambridge University Press. 2008. Retrieved on 30 September from <http://www.nlp.stanford.edu/IR-book/>
- [6] Bogatyy I. (2016) "Predicting answer types for question-answering". Retrieved on 30 September from <https://cs224d.stanford.edu/reports/Bogatyy.pdf>
- [7] 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), 981-997. Retrieved on 17 September 2016 from <https://soar.smu.github.io/papers/jcst-duplicateqns.pdf>. doi:10.1007/s11390-015-1576-4
- [8] 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. Retrieved on 12 October from [www.jmlr.org/papers/volume3/blei03a/blei03a.pdf](http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf)
- [9] Hoffman M. D., Blei D. M. and Bach F. (2010). "Online Learning for Latent Dirichlet Allocation" Retrieved on 12 October from <http://www.cs.princeton.edu/~blei/papers/HoffmanBleiBach2010b.pdf>
- [10] Chopra S., Hadsell, R. and LeCun, Y. (2005). "Learning a Similarity Metric Discriminatively, with Application to Face Verification" Retrieved on 17 November from <http://yann.lecun.com/exdb/publis/pdf/chopra-05.pdf>
- [11] Nallapati R., Zhou B., Nogueira dos santos C., Gulcehre C. and Xiang B. (2016). "Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond" Retrieved on 23 November from <https://arxiv.org/abs/1602.06023>
- [12] Elith J. and Leathwick J. (2016). "Boosted Regression Trees for ecological modelling" Retrieved on 24 November from <https://cran.r-project.org/web/packages/dismo/vignettes/brt.pdf>
- [13] Yassir A., and Nayak S. (2012) "A Reliable Effective Terascale Linear Learning System". In *International Journal of Computer Science and Communication Networks*, 2(1), 93-98. Retrieved on 24 November from <http://www.ijcsn.com/Documents/Volumes/vol2issue1/ijcsn2012020116.pdf>
- [14] Reddit OMSCS FAQ. (n.d.). Retrieved December 03, 2016, from <https://www.reddit.com/r/OMSCS/wiki/index>
- [15] Agarwal A., Chapelle O., Dudik M. and Langford J. (2012). "A Reliable Effective Terascale Linear Learning System". In *Journal of Machine Learning Research* 2012. Retrieved on 24 November from <https://arxiv.org/pdf/1110.4198v3.pdf>
- [16] Saffari, A., Leistner, C., Santner, J., Godec, M., and Bischof, H. (2009). "On-line Random Forests". 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops. doi:10.1109/iccvw.2009.5457447. Retrieved on 3 December from <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.150.1671&rep=rep1&type=pdf>
- [17] Lin CY. (2004). "ROUGE: A Package for Automatic Evaluation of Summaries". In *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*, Barcelona, Spain, July 25 - 26, 2004. Retrieved on 24 November from <http://www.aclweb.org/anthology/W/W04/W04-1013.pdf>