Addressing Duplicate Questions in Stack Overflow:An Empirical Study

Vartika Agrahari
Indian Institute of Technology
Tirupati, India
cs18m016@iittp.ac.in

ABSTRACT

Stack Overflow is a world-wide question-answer platform primarily for programming challenges. Despite efforts to prevent repeated questions,site contains a lot of duplicate questions,thus causing users to unwantedly wait for getting answers of those questions,which already have been answered or asked. Currently the site relies on high reputed developers and moderators to manually mark the duplicate questions,thus causing a lot of time and additional efforts.They came up with classification method,based on manual examination,which uses a number of carefully selected features to check the duplicate questions.Also,they took a comparison of the proposed technique with existing state-of-the-art method called as DupPredictor,and found a better recall-rate.

CCS CONCEPTS

• **Information systems → Social networking sites**;

KEYWORDS

Stack Overflow,duplicate questions,classifier

ACM Reference Format:
Vartika Agrahari. 2018. Addressing Duplicate Questions in Stack Overflow:An Empirical Study. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

WEEK 1

1 LITERATURE REVIEW

While addressing this problem ,I went to a number of other related researches.First I saw a paper called *Finding Near-Duplicate Web Pages: A Large-Scale Evaluation of Algorithms* where they used Broder's shingling algorithm and Charikar's random projection based approach which were said to be state-of-the-art algorithms used for getting near-duplicate pages. They estimated the two algorithm on a very large set of 1.6B different web pages which showed that neither of them were finding near-duplicate pages on same site,but were good in finding the same on different site.Finally they found that Charikar's algorithm gets more near-duplicate pairs on different sites and yields a better accuracy taking everything in account, namely 0.50 vs 0.38 for Broder's algorithm. At last ,they gave a combined accuracy figure of 0.79 or 79%.

Secondly,I saw paper named *How do programmers ask and answer questions on the web? (NIER track)* which depicted the human question-asking behaviour analysis. Getting an idea of these question-answer websites and understanding the crux behind it,helps the companies to benefit the maximum they can.In this paper, they analyzed the Stack Overflow data,and tried to find an estimate of what type of questions are asked on this website ,and what kind of questions are been answered. They primarily found that these websites are useful in reviewing questions related to coding challenges and conceptual behaviour. They constructed research questions and gave a view of future work they can help developers do to make these sites more and more useful, and to perceive the implication of turning these QA exchanges into technical mini-blogs through the editing of questions and answers.

Then,I went through a paper called *Adaptive Near-Duplicate Detection via Similarity Learning* where they came up with a typical near-duplicate detection mechanism which can be adapted in any particular domain and area. They took each document as a real-valued sparse k-gram vector, where we optimize weights using a specified similarity check measure,say cosine similarity or the Jaccard coefficient. Near-duplicate pages can be easily detected through this enhanced and improved similarity measure. Additionally ,for more efficient similarity calculation,we can map these vectors to a small number of hash-values as document signatures using the locality sensitive hashing scheme. They illustrated their technique in two domains: Web news articles and email messages.Their approach was not only more accurate in comparison to the algorithms such as Shingles and I-match,but demonstrated a continuous enhancement in the particular domain, which was the desired feature lacking in the previous methods.

I also gone through a paper named *Harnessing Stack Overflow for the IDE* where they came up with an Eclipse plugin named , Seahawk which consolidates the Stack Overflow crowd knowledge in the IDE .This results in the smooth access of Stack Overflow data,thus getting answers to the questions,without worrying about context switching. They introduced their primary conclusions on Seahawk as:It allows users to (1) retrieve and access QA from Stack Overflow, (2) link the related conversations to any source code in Eclipse, and (3) provide comments to the link for explanation.

The publication *An Empirical Study on Developer Interactions in StackOverflow* analyzed the usage of tags in Stack Overflow. In this paper,they further extended their study,to get the deeper insight on the interaction of developers with one another on these QA sites.Firstly,they took a observation on the distribution of developers who ask and answers questions on these sites.Also,they

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

analyzed if there is any division of the users into questioners and answerers .Further ,they performed automated text-mining to get various kinds of subjects or topics asked by the users. They utilized a well known topic modelling method,called as Latent Dirichlet Allocation(LDA),to examine tens of thousand of QA, and yielded five topics from them. This method of topic-modelling provided a distinct perception in comparison to the other researches for categorization of topics in Stack Overflow. Now each and every question can be labelled into different topics with distinct probabilities, Each question can now be categorized into several topics with different probabilities, and the new learned topic model can automatically tag a new question to different categories having varying probabilities. Lastly, they also depicted the distribution of questions and developers belonging to number of different topics resulted by LDA.

2 RESEARCH QUESTIONS

Typically,a number of questions have been asked:
(1) What is the reason behind the developers duplicating questions in Stack Overflow?
To answer this question,first of all we need to educate the developers so that they ask questions in effective manner,and for that we need to understand why do they duplicate questions.This will help the moderators to take the right step to prevent the variance of duplication.
(2) Can an automated method be developed from machine learning perspective for avoiding and detecting replicated questions?
Here,they tried to develop a classification approach with various number of carefully selected feature to help us in detecting the replicated copies of the questions.
(3) In comparison to other duplicate text detection technique, how this tool performs better and with more accuracy?
They compared their tool with the *Dup preditor* which gives better recall-rates.
(4)Can previous work be related and extended in this paper?
There has been a considerable amount of work in different aspect of Stack Overflow which includes question kinds, patterns of user interaction, women engagement , code examples analysis, topic division , web relevant conversations, question-asking behaviour of users , and developers distribution in asking-answering questions,but none of them focused on duplicate questions problem.
(5)Are duplicate questions similar to duplicate bug reports?
There are primarily two fields,say textual and non-textual information in bug reports,which can not be found in Stack Overflow questions. Also,unlike questions in Stack Overflow having tags, bug reports don't have tags. Thus because of these differences in both,it is worthwhile to take duplicate questions Stack Overflow into consideration for further examination. .
(6)What is the reputation of users who ask duplicate questions?
In desire of getting answers immediately and without any wait or search,users having least experience tend to ask duplicate questions as they don't want to review the site. Following them comes the developers with minimal experience showing a little interest in searching and reviewing the site.However,the users with mid reputation tend to ask only 25% of the repeated questions. Developers with high reputation and a lot of experience are generally more informed and don't tend to ask duplicate questions..
(7)Why user tend to ask duplicate questions?
They gave a number of reasons for this:
1.Avoiding initial searches in Stack Overflow.
2.Titles do not match.
3.Despite task similarity,if there is domain difference.
4.Ambiguous and descriptive to grasp.
5.Too brief to get proper understanding.
6.Lack of knowledge about the particular problem.
7.Not having proper idea of terminology/buzzwords .
(8)What is the Proposed Technique?
Basically, the method is divided into three phases.During first process,the pre-process the text and get it ready for feature extraction process . In second phase,to generate the binary classification model, they do different feature collection for each pair of questions. In the last phase , duplicate question are detected by utilizing the previously trained classifier to get a ranked list of results.

3 TOOLS USED TO EXECUTE STUDY

1.In the first paper I mentioned,they used Broder's shingling algorithm and Charikar's random projection algorithm.
2.The second publication I saw was a analytical research and they manually tried to find the answers of mainly two questions:
(i)What are the different kinds of questions asked for programmers on these question-answers websites?
(ii)What is the proportion of questions being answered and unanswered?
3.In the third publication,they took each document as a real-valued sparse k-gram vector, where they optimize weights using a specified similarity check measure,say cosine similarity or the Jaccard coefficient..
4.I the fourth publication,they came up with a tool callled Seahawk, an Eclipse plugin to consolidate the Stack Overflow crowd knowledge in the IDE.
5.In the fifth paper,they used a popular topic modelling method,Latent Dirichlet Allocation (LDA), to observe insight of tens of thousands of questions and answers, and generate five relevant topics.

4 DATA SETS USED

1.The paper *Finding Near-Duplicate Web Pages: A Large-Scale Evaluation of Algorithms* took a comparison of two algorithms, on the large data-set of 1.6B distinct web pages.
2. The paper *How Do Programmers Ask and Answer Questions on the Web? (NIER Track)* analysed 38,419 questions 31,729 owners, 68,467 answers and 111,408 tag instances.
3.The paper *Adaptive Near-Duplicate Detection via Similarity Learning* demonstrated their method in two domains: Email messages and web news articles.
4.The paper *Harnessing Stack Overflow for the IDE* extracted Stack Overflow data.
5.The paper *An Empirical Study on Developer Interactions in Stack*

Overflow using Stack Overflow API,extracted the first 100,000 questions. .They can download at most 30,000 questions from an IP address in a day. 100 randomly (or pseudo-randomly) selected questions are returned by API for every call. Thus,few of the 100,000 questions that are returned are duplicates of another such they are having the same question identifier.In total they got 63,863 unique questions by applying filter.

5 UNSOLVED PROBLEM IN THIS AREA

1.Instead of considering major programming languages,we can give emphasis on all languages and their duplication.

2.Can be thought of as implementing a web service so that not only Stack Overflow,but other sites can utilize it.

3.While user posting a duplicate question,we can generate a pop up that this question has already been asked,thus avoids duplication and saves the time of developers.

4.How can accuracy be increased further to detect duplicate questions?

5.We can use deep learning concepts to get better recall-rates and accuracy.

6 REFERENCES

[1]Monika Henzinger, Finding near-duplicate web pages: a large-scale evaluation of algorithms, Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, August 06-11, 2006, Seattle, Washington, USA [doi>10.1145/1148170.1148222]

[2]Christoph Treude , Ohad Barzilay , Margaret-Anne Storey, How do programmers ask and answer questions on the web? (NIER track), Proceedings of the 33rd International Conference on Software Engineering, May 21-28, 2011, Waikiki, Honolulu, HI, USA [doi>10.1145/1985793.1985907]

[3]Hannaneh Hajishirzi , Wen-tau Yih , Aleksander Kolcz, Adaptive near-duplicate detection via similarity learning, Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval, July 19-23, 2010, Geneva, Switzerland [doi>10.1145/1835449.1835520]

[4]A. Bacchelli, L. Ponzanelli and M. Lanza, âĂIJHarnessing Stack Overflow for the IDEâĂİ, In Proc. of RSSE, 2012, pp. 26-30.

[5]Shaowei Wang , David Lo , Lingxiao Jiang, An empirical study on developer interactions in StackOverflow, Proceedings of the 28th Annual ACM Symposium on Applied Computing, March 18-22, 2013, Coimbra, Portugal [doi>10.1145/2480362.2480557]

WEEK 2

7 DATASET CREATION

7.1 Data source

Stack Overflow

7.2 Dataset

The dataset includes 8000 questions on total 8 tags namely Java, Javascript,Strings,Python,Arrays,SQL, Android and C++.So total 8000 questions of Stack Overflow of 8 different tags has been extracted. Each row of each table in the dataset attached contains the title of the question,its url and its body content and the questions are already separated on the basis of tags.Thus,for each question we have its title,body content,url and its tag.

7.3 Process of Extraction

We scraped the question,by the help of standard python library called as csv,requests,BeautifulSoup. With the help of requests library,we are getting to the website ,basically the get function allows us to do that.Next,by using BeautifulSoup we are scraping the question from the site,and lastly by using the csv library the extracted data is converted into comma separated values(csv)format and stored in .csv file. Now for managing database,we are using the SQLite Database Manager.Above extracted csv file is imported into SQLite.Thus in our database total 8 tables are there for 8 different tags.

7.4 Revised Research Questions

(1) Can an automated method be developed from machine learning perspective for avoiding and detecting replicated questions?

Here,we are trying to build a tool to detect the duplicate question with higher accuracy by taking the above dataset.

(2)What new can be done to increase the accuracy in comparison to existing tools?

We can come up with the techniques better than the existing technique involving concepts of deep learning.

7.5 Next Step to be done

Now we need to refine the dataset by dividing the dataset into duplicate questions and their master questions,so that we can give it to our classification models. And also we need to take random test dataset for testing the classifier.

Week 3

8 IDEA OF THE STRUCTURE

Now we are first trying to create a prototype of what has to be done.Idea of the structure is: 1.Refine the dataset to create a training dataset to create the classifier we need. 2.In the training dataset we take pair of master and duplicate question in each record along with its tag,title,body-content of the question. 3.Using this to calculate a number of feature values namely,cosine similarity value,term-overlap measure,entity overlap,entity-type overlap,semantic similarity for training the classifier. 4.Now here we are using the logistic regression as the machine learning technique to predict whether a question is duplicate or not. 5.After training the classifier,we need to test some sample random question and analyze the accuracy.

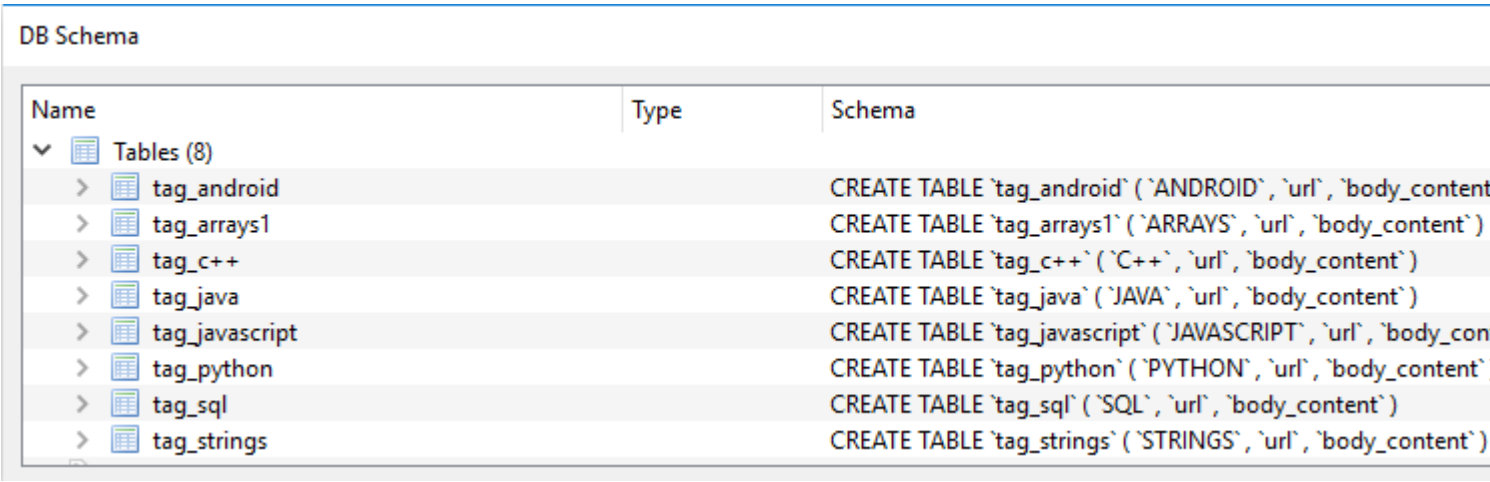

Figure 1: Figure showing schema of the extracted data.

9 TECHNIQUE USED

To conduct the experiment we are using Logistic Regression to train the classifier.It is a binary classifier as thus satisfies our requirements of only two classes,namely duplicate and non-duplicate.

10 EXPECTED RESULT

The expectation is to get better than result than the pre-existing technique DUPE.
Also to further increase the accuracy,we can go for deep learning models.

11 RESEARCH QUESTIONS

(1) Can an automated method be developed from machinelearning perspective for avoiding and detecting replicatedquestions?
Here,we are trying to build a tool to detect the duplicate question-with higher accuracy by taking the above dataset.
(2)What new can be done to increase the accuracy in com-parison to existing tools?
We can come up with the techniques better than the existing technique involving concepts of deep learning.

WEEK 4

12 CONDUCTING THE EXPERIMENT

Before conducting the whole experiment,we are first trying to create a prototype of what has to be done. Thus we are considering a small dataset for meanwhile and later will expand it by refining the extracted data manually.
Right now,we are considering about 10 questions in training dataset and storing this database in sqlite3. Below are the implementation details:
1.Connecting to sqlite3 to get the database.
2.Pre-processing the data,for which we used Porter's Stemming algorithm for stemming and nltk to remove the stopwords and tokenize the text.
3.For calculating the first feature,we are using cosine similarity value.For getting this value we are using a number of libraries say numpy,scipy,pandas for matrix operations,handling the dataframe etc.
4.Next we calculate the second feature called as term overlap.It is basically a text similarity function in normalised form between 0 and 1. It is calculated by 2*(|t1âĹĭt2|)/(|t1|âĹł|t2|).
5.Thirdly,we calculate the Entity Overlap by using the Standford Name Entity Recognition(NER).
6.Likewise,we calculate the next two features entity-type overlap and word-net similarity(yet to be done
7.For all the features being calculated we are storing it in .csv file to make training dataset.

13 NEXT STEP TO BE DONE

1.After calculating all the feature values,we need to train the classifier using logistic regression.
2.Manually refine the dataset to create a large dataset for our final experiment.
3.Calculate the accuracy by giving test random samples.

WEEK 5

1.Here,I tried to extract the other two features.Thus total 5 features are there to train the classifier. I also tried to increase the dataset.
2.I am facing little difficulty in scraping the data in the way it needed to build classifier.As we need only first duplicate questions and non-duplicate questions in equal proportion to train the classifier.Thus it is taking time to clean the data I scraped.
3.After extracting all the features and building the training dataset,we need to build the classifier.
4.I will use logistic regression to build the classifier.