Automatic Duplicate Question Detection in Stack over Flow

Dr.D.Anusha[1], S.Prabhat Kumar[2], M.Uday Kiran Reddy[3], CH.Akash[4], Y.Pavan Sai[5]

[1]Associate Professor, Department of CSE-Artificial Intelligence and Machine Learning, S.R.K Institute of Technology, NTR, Andhra Pradesh, India, itsmeprabhatkumar4u@gmail.com.
[2] student, Department of CSE-Artificial Intelligence and Machine Learning, S.R.K Institute of Technology, NTR, Andhra Pradesh, India
[3] student, Department of CSE- CSE-Artificial Intelligence and Machine Learning, S.R.K Institute of Technology, NTR, Andhra Pradesh, India
[4] student, Department of CSE- Artificial Intelligence and Machine Learning, S.R.K Institute of Technology, NTR, Andhra Pradesh, India
[5] student, Department of CSE- Artificial Intelligence and Machine Learning, S.R.K Institute of Technology, NTR, Andhra Pradesh, India

Abstract:

Stack Overflow is a popular Community-based Question Answer (CQA) website focused on software programming and has attracted more and more users in recent years. However, duplicate questions frequently appear in Stack Overflow and they are manually marked by the users with high reputation. Automatic duplicate question detection alleviates labor and effort for users with high reputation. Although existing approaches extract textual features to automatically detect duplicate questions, these approaches are limited since semantic information could be lost. To tackle this problem, we explore the use of powerful deep learning techniques, including Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM), to detect duplicate questions in Stack Overflow. In addition, we use Word2Vec to obtain the vector representations of words. They can fully capture semantic information at the document level and word level respectively. Therefore, we construct three deep learning approaches WV- CNN, WV-RNN and WV-LSTM, which are based on Word2Vec, CNN, RNN and LSTM, to detect duplicate questions in Stack Overflow. Evaluation results show that WV-CNN and WV-LSTM have made Significant improvements over four baseline approaches (i.e., Duplicate Predictor, Dupe, Duplicate Predictor Repeat, and Duplicate Repeat) and three deep learning approaches (i.e., DQ-CNN, DQ-RNN, and DQ-LSTM) in terms of recall-rate@5, recall- rate@10 and recall-rate@20. Furthermore, the experimental results indicate that our approaches WV-CNN, WV-RNN, and WV-LSTM outperform four machine learning approaches based on Support Vector Machine, Logic Regression, Random Forest and extreme Gradient Boosting in terms of recall-rate@5, recall-rate@10 and recall-rate@20

Key Words: Convolutional Neural Networks (CNN), recall-rate@5, recall- rate@10. recall-rate@20, Stack Overflow Dataset, LDA, Flask, Tkinter.

1. Introduction:

Duplicate questions make Stack Overflow site maintenance harder, waste resources that could have been used to answer other questions and cause developers to unnecessarily wait for answers that are already available. A typical question in Stack Overflow contains several fields, such as submitter, title, description, tags, and comments. +A developer needs to provide all three pieces of information when he/she submits a question to Stack Overflow. The title is a summary of the question, the description is a detailed explanation of the question, and tags are sets of words or short phrases that capture important aspects of the question. The goal is to implement Duplicate Predictor which takes input from a new question and gives output in the form of **k** duplicate questions as the output by considering multiple factors.

2. Existing System:

[1] Search Functionality: Users can search for existing questions before posting their own. Stack Overflow's search functionality is quite robust and often returns relevant results, which can help users avoid asking duplicate questions.

[2] Related Questions: When asking a new question, Stack Overflow automatically suggests related questions based on the title and content of the question being asked. This can help users identify potential duplicates before posting.

[3] Community Moderation: Stack Overflow relies heavily on its community of users to moderate content. If a question is identified as a duplicate, community members can vote to close it as such. Additionally, users with sufficient reputation can mark questions as duplicates, which helps organize content and improve the overall quality of the platform.

[4] Flagging: Users can flag questions that they believe are duplicates. Moderators can then review these flags and take appropriate action if necessary.

[5] Data Analysis: Stack Overflow has access to a vast amount of data regarding questions and answers. They may use this data to identify patterns and potentially automate the process of detecting duplicate questions in the future. However, as of my last update, I'm not aware of any specific system or tool developed by Stack Overflow for this purpose.

3. Literature Survey:

1. D. Correa and A. Sureka, "Chaff from the wheat: Characterization and modeling of deleted questions on stack overflow", *Proc. 23rd Int. Conf. World Wide Web (WWW)*, pp. 631-642, Jan. 2014.

Stack Overflow is the most popular Community based Question Answering (CQA) website for programmers on the web with 2.05M users, 5.1M questions and 9.4M answers. Stack Overflow has explicit, detailed guidelines on how to post questions and an ebullient moderation community. Despite these precise communications and safeguards, questions posted on Stack Overflow can be extremely off topic or very poor in quality. Such questions can be deleted from Stack Overflow at the discretion of experienced community members and moderators. We present the first study of deleted questions on Stack Overflow. We divide our study into two parts - (i) Characterization of deleted questions over ~5 years (2008-2013) of data, (ii) Prediction of deletion at the time of question creation. Our characterization study reveals multiple insights on question deletion phenomena.

Stack Overflow has become a fundamental element of developer toolset. Such influence increase has been accompanied by an effort from Stack Overflow community to keep the quality of its content. One of the problems which jeopardizes that quality is the continuous growth of duplicated questions. To solve this problem, prior works focused on automatically detecting duplicated questions. Two important solutions are DupPredictor and Dupe. Despite reporting significant results, both works do not provide their implementations publicly available, hindering subsequent works in scientific literature which rely on them. We executed an empirical study as a reproduction of DupPredictor and Dupe. Our results, not robust when attempted with different set of tools and data sets, show that the barriers to reproduce these approaches are high. Furthermore, when applied to more recent data, we observe a performance decay of our both reproductions in terms of recall-rate over time, as the number of questions increases. Our findings suggest that the subsequent works concerning detection of duplicated questions in Question-and-Answer communities require more investigation to assert their findings.

2. Multi-Factor Duplicate Question Detection in Stack Overflow Y. Zhang, D. Lo, X. Xia, et al. Journal of Computer Science and Technology, 2015

In their paper, Zhang et al. propose an automated approach named Dup Predictor for detecting duplicate questions on Stack Overflow. The approach considers multiple factors including title, description, latent topics, and tags of the questions. They conduct experiments on a dataset containing over two million questions, achieving a recall-rate@20 score of 63.8%. Compared to the standard search engine of Stack Overflow, Dup Predictor improves the recall-rate@10 score by 40.63%. Furthermore, it outperforms other approaches, demonstrating its effectiveness in detecting duplicate questions and enhancing the quality of Stack Overflow. Stack Overflow has become a fundamental element of developer toolset. Such influence increase has been accompanied by an effort from Stack Overflow community to keep the quality of its content. One of the problems which jeopardizes that quality is the continuous growth of duplicated questions.

To solve this problem, prior works focused on automatically detecting duplicated questions. Two important solutions are Dup Predictor and Dupe. Despite reporting significant results, both works do not provide their implementations publicly available, hindering subsequent works in scientific literature which rely on them. We executed an empirical study as a reproduction of Dup Predictor and Dupe. Our results, not robust when attempted with different set of tools and data sets, show that the barriers to reproduce these approaches are high. Furthermore, when applied to more recent data, we observe a performance decay of our both reproductions in terms of recall-rate over time, as the number of questions increases. Our findings suggest that the subsequent works concerning detection of duplicated questions in Question-and-Answer communities require more investigation to assert their findings.

3. M. Ahasanuzzaman, M. Asaduzzaman, C. K. Roy and K. A. Schneider, "Mining duplicate questions in stack overflow", *Proc. 13th Min. Software. Repositories (MSR)*, pp. 402-412, May 2016.

Stack Overflow is a popular question answering site that is focused on programming problems. Despite efforts to prevent asking questions that have already been answered, the site contains duplicate questions. This may cause developers to unnecessarily wait for a question to be answered when it has already been asked and answered. The site currently depends on its moderators and users with high reputation to manually mark those questions as duplicates, which not only results in delayed responses but also requires additional efforts. Among the numerous questions posted in Stack Overflow, two or more of them may express the same point and thus are duplicates of one another.

Duplicate questions make Stack Overflow site maintenance harder, waste resources that could have been used to answer other questions, and cause developers to unnecessarily wait for answers that are already available. To reduce the problem of duplicate questions, Stack Overflow allows questions to be manually marked as duplicates of others. Since there are thousands of questions submitted to Stack Overflow every day, manually identifying duplicate questions is a difficult work. Thus, there is a need for an automated approach that can help in detecting these duplicate questions.

4. R. F. Silva, K. Paixão and M. de Almeida Maia, "Duplicate question detection in stack overflow: A reproducibility study", *Proc. 25th Software. An. Ev. Reeng. (SANER)*, pp. 572-581, Mar. 2018.

Stack Overflow has become a fundamental element of developer toolset. Such influence increase has been accompanied by an effort from Stack Overflow community to keep the quality of its content. One of the problems which jeopardizes that quality is the continuous growth of duplicated questions. To solve this problem, prior works focused on automatically detecting duplicated questions. Two important solutions are Dup-Predictor and Dupe. Despite reporting significant results, both works do not provide their implementations publicly available, hindering subsequent works in scientific literature which rely on them. We executed an empirical study as a reproduction of Dup-Predictor and Dupe. Our results, not robust when attempted with different set of tools and data sets, show that the barriers to reproduce these approaches are high.

Furthermore, when applied to more recent data, we observe a performance decay of our both reproductions in terms of recall-rate over time, as the number of questions increases. Our findings suggest that the subsequent works concerning detection of duplicated questions in Question-and-Answer communities require more investigation to assert their findings. Among the numerous questions posted in Stack Overflow, two or more of them may express the same point and thus are duplicates of one another. Duplicate questions make Stack Overflow site maintenance harder, waste resources that could have been used to answer other questions, and cause developers to unnecessarily wait for answers that are already available. To reduce the problem of duplicate questions, Stack Overflow allows questions to be manually marked as duplicates of others. Since there are thousands of questions submitted to Stack Overflow every day, manually identifying duplicate questions is a difficult work.

5. C. N. Kamath, S. S. Bukhari and A. Dengel, "Comparative study between traditional machine learning and deep learning approaches for text classification", *Proc. ACM Symp. Document Eng.*, pp. 1-11, 2018.

In this contemporaneous world, it is an obligation for any organization working with documents to end up with the insipid task of classifying truckload of documents, which is the nascent stage of venturing into the realm of information retrieval and data mining. But classification of such humongous documents into multiple classes, calls for a lot of time and labor. Hence a system which could classify these documents with acceptable accuracy would be of an unfathomable help in document engineering. We have created multiple classifiers for document classification and compared their accuracy on raw and processed data. We have garnered data used in a corporate organization as well as publicly available data for comparison. Duplicate question detection is a crucial aspect of maintaining the quality and usability of online Q&A platforms like Stack Overflow. In this review, we delve into the mechanisms and strategies employed by Stack Overflow to effectively identify duplicate questions.

Through a combination of natural language processing techniques, machine learning algorithms, community moderation, and metadata analysis, Stack Overflow has developed a robust system capable of accurately detecting duplicate questions. This review explores the various components of Stack Overflow's duplicate question detection system, highlighting its strengths, limitations, and potential areas for improvement. Stack Overflow is a prominent online community for programmers seeking answers to their coding queries. With millions of questions and answers spanning various programming languages and technologies, the platform faces the challenge of preventing the proliferation of duplicate questions. Duplicate questions not only clutter the platform but also hinder the discovery of relevant information for users. To address this challenge, Stack Overflow has implemented a sophisticated duplicate question detection system, which we aim to dissect and evaluate in this review.

6. J. Kapočiūtė-Dzikienė, R. Damaševičius and M. Wozniak, "Sentiment analysis of Lithuanian texts using traditional and deep learning approaches", *Computers*, vol. 8, no. 1, pp. 1-16, Jan. 2019.

We describe the sentiment analysis experiments that were performed on the Lithuanian Internet comment dataset using traditional machine learning (Naïve Bayes Multinomial—NBM and Support Vector Machine—SVM) and deep learning (Long Short-Term Memory—LSTM and Convolutional Neural Network—CNN) approaches. The traditional machine learning techniques were used with the features based on the lexical, morphological, and character information. The deep learning approaches were applied on the top of two types of word embeddings (*Vord2Vec* continuous bag-of-words with negative sampling and *FastText*). Both traditional and deep learning approaches had to solve the positive/negative/neutral sentiment classification task on the balanced and full dataset versions. The best deep learning results (reaching 0.706 of accuracy) were achieved on the full dataset with CNN applied on top of the *FastText* embeddings, replaced emoticons, and eliminated diacritics. The traditional machine learning approaches demonstrated the best performance (0.735 of accuracy) on the full dataset with the NBM method, replaced emoticons, restored diacritics, and lemma unigrams as features.

Although traditional machine learning approaches were superior when compared to the deep learning methods; deep learning demonstrated good results when applied on the small datasets. As the platform continues to evolve, ongoing efforts to enhance the efficiency and accuracy of the duplicate question detection system will be essential in maintaining its integrity and usefulness. Nonetheless, there is room for refinement, particularly in addressing edge cases and enhancing the scalability and real-time responsiveness of the system. Future advancements may involve exploring more sophisticated deep learning architectures, refining feature engineering strategies, and further integrating user feedback mechanisms. In conclusion, Stack Overflow's duplicate question detection system stands as a testament to the power of combining automated algorithms with community-driven moderation.

4. Proposed System:

[1] Data Collection: Obtain a dataset of questions from Stack Overflow, including metadata such as question titles, bodies, tags, timestamps, and accepted answers. Ensure the dataset is diverse and represents various programming languages, technologies, and topics.

[2] Preprocessing: Clean and preprocess the text data by removing HTML tags, code snippets, punctuation, and stop words. Tokenize the text into words or phrases and convert them to lowercase. Apply stemming or lemmatization to reduce words to their base forms.

[3] Feature Extraction: Extract features from the pre-processed text data that capture semantic similarity between questions.

[4] Common techniques include Bag-of-words (BoW): Represent each question as a vector of word frequencies.

[5] TF-IDF (Term Frequency-Inverse Document Frequency): Weight words based on their importance in the document and across the entire corpus.

[6] Word embeddings: Use pre-trained word embeddings (e.g., Word2Vec, GloVe) to represent words as dense vectors in a high-dimensional space.

[7] Doc2Vec: Represent entire questions as fixed-length vectors using techniques like Paragraph Vector.

[8] Model Training: Train a machine learning or deep learning model to predict whether a pair of questions are duplicates.

[9] Common model architectures include:

Siamese networks: Learn embeddings for question pairs and measure their similarity.
Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs) for sequence modelling. Transformer-based architectures like BERT or Roberta fine-tuned for duplicate detection. Use a labelled dataset for training, where duplicate question pairs are labelled as such.

[10] Evaluation: Evaluate the trained model on a separate test dataset using metrics such as accuracy, precision, recall, F1-score, and ROC-AUC. Perform cross-validation to ensure the model's generalization ability.

[11] Integration with Stack Overflow: Develop an API or service that accepts a new question as input and returns a list of potential duplicate questions from the Stack Overflow database.

Implement mechanisms to handle large-scale querying and response times efficiently. Ensure the system complies with Stack Overflow's terms of service and API usage policies.

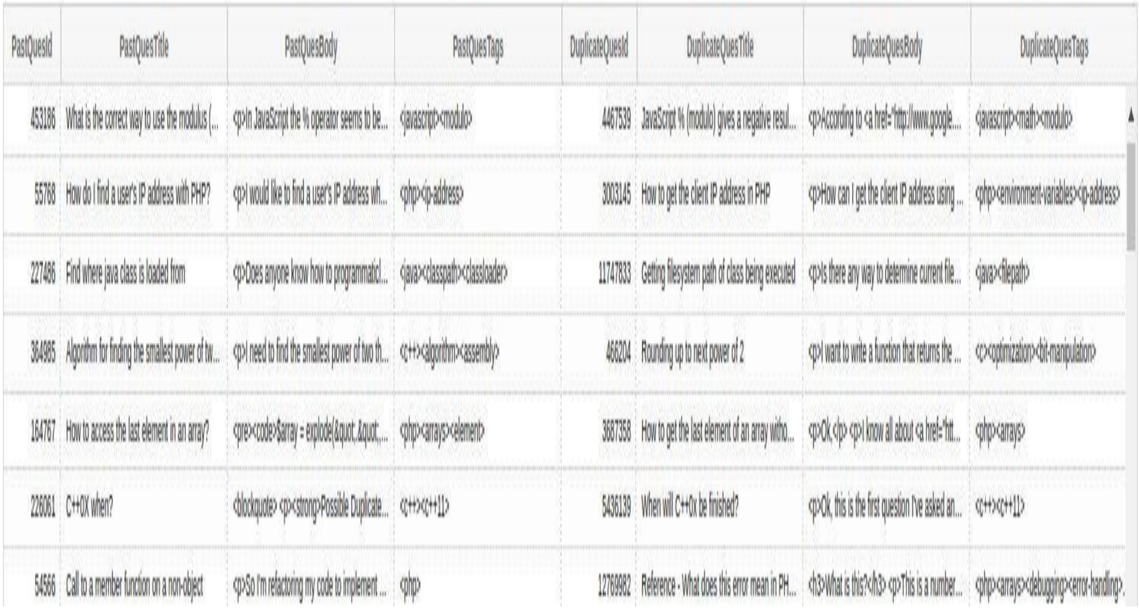
[12] Deployment and Monitoring: Deploy the duplicate question detection system in a production environment. Monitor the system's performance and periodically retrain the model with new data to adapt to changing trends and user behaviour. Collect feedback from users and moderators to continuously improve the system.

[13] Maintenance and Updates: Regularly update the system with the latest techniques and improvements in natural language processing and machine learning. Stay informed about changes to the Stack Overflow platform and adapt the system accordingly.

5. Dataset:

To build a duplicate question detection system using a dataset from Stack Exchange (which includes Stack Overflow), you can utilize the Stack Exchange Data Dump. Stack Exchange provides anonymized dumps of its data, including questions, answers, user information, and more, which you can download and use for research purposes. Stack Exchange periodically releases data dumps on the Internet Archive. You can visit the following link to download the latest dumps:

- https://data.stackexchange.com/stackoverflow/queries
- It has around 50k entries which includes previous question (Id, Title, Description, Tags) with Duplicate Question (Id, Title, Description, Tags).



PastQuesId	PastQuesTitle	PastQuesBody	PastQuesTags	DuplicateQuesId	DuplicateQuesTitle	DuplicateQuesBody	DuplicateQuesTags
453186	What is the correct way to use the modulus (...	<p>In JavaScript the % operator seems to be...	<javascript><modulo>	4467539	JavaScript % (modulo) gives a negative resul...	<p>According to <a href="http://www.google...	<javascript><math><modulo>
55768	How do I find a user's IP address with PHP?	<p>I would like to find a user's IP address wh...	<php><ip-address>	3003145	How to get the client IP address in PHP	<p>How can I get the client IP address using ...	<php><environment-variables><ip-address>
227486	Find where java class is loaded from	<p>Does anyone know how to programmatic...	<java><classpath><classloader>	11747833	Getting filesystem path of class being executed	<p>Is there any way to determine current file...	<java><filepath>
364985	Algorithm for finding the smallest power of tw...	<p>I need to find the smallest power of two th...	<c++><algorithm><assembly>	466204	Rounding up to next power of 2	<p>I want to write a function that returns the ...	<c><optimization><bit-manipulation>
164767	How to access the last element in an array?	<pre><code>$array = explode(&quot;,&quot;,...	<php><arrays><element>	3687358	How to get the last element of an array witho...	<p>Ok.</p> <p>I know all about <a href="htt...	<php><arrays>
226061	C++0X when?	<blockquote> <p><strong>Possible Duplicate...	<c++><c++11>	5436139	When will C++0x be finished?	<p>Ok, this is the first question I've asked an...	<c++><c++11>
54566	Call to a member function on a non-object	<p>So I'm refactoring my code to implement ...	<php>	12769982	Reference - What does this error mean in PH...	<h3>What is this?</h3> <p>This is a number...	<php><arrays><debugging><error-handling>.

Fig (1) Datasets of Stack Exchange

Select the Stack Exchange Sites: Stack Exchange hosts multiple Q&A sites on various topics. You can choose to download dumps for specific sites or the entire network. For duplicate question detection on Stack Overflow, you'll primarily focus on the Stack Overflow dataset.

Explore the Data Schema: The data dump is provided in XML format, and it contains tables representing different entities such as questions, answers, users, tags, etc. Before proceeding, familiarize yourself with the schema and understand the structure of the data.

Import Data into a Database: Once you've downloaded the data dump, you can import it into a relational database management system (RDBMS) like MySQL, PostgreSQL, or SQLite. This allows you to query and manipulate the data efficiently.

Preprocess and Extract Features: Preprocess the text data by cleaning and tokenizing question titles and bodies. Extract relevant features such as TF-IDF vectors, word embeddings, or other representations that capture semantic similarity between questions.

Label Duplicate Question Pairs: Annotate pairs of questions in the dataset as either duplicates or non-duplicates. You can use existing duplicate question links provided in the dataset or implement a labelling mechanism based on similarity thresholds.
Split Data for Training and Testing: Split the annotated dataset into training and testing sets for model development and evaluation. Ensure that both sets contain a balanced distribution of duplicate and non-duplicate question pairs.
Train and Evaluate Model: Train a duplicate question detection model using machine learning or deep learning algorithms. Evaluate the model's performance using appropriate evaluation metrics such as accuracy, precision, recall, F1-score, etc., on the test set.
Integrate with Stack Overflow: Develop an application or service that integrates the trained model to detect duplicate questions on Stack Overflow in real-time. You can utilize the Stack Exchange API to fetch new questions and provide duplicate suggestions to users.

6. Methodology:

[1] Data Acquisition: Download the Stack Exchange Data Dump from the provided link. Choose the Stack Exchange sites relevant to your project, primarily focusing on Stack Overflow. Extract the relevant data from the downloaded files.
[2] Data Exploration: Understand the schema of the dataset and the structure of tables. Explore the attributes available in the question table, such as question titles, bodies, tags, timestamps, etc. Get a sense of the volume of data available and its distribution.
[3] Data Preprocessing: Clean the text data by removing HTML tags, code snippets, and other noise. Tokenize the text into words or phrases. Convert text to lowercase to ensure consistency. Apply techniques like stemming or lemmatization to reduce words to their base forms. Handle missing values and outliers appropriately.
[4] Feature Engineering: Extract features from the pre-processed text data that capture semantic similarity between questions. Common features include TF-IDF vectors, word embeddings (Word2Vec, GloVe), and document embeddings (Doc2Vec). Experiment with different feature representations to find the most effective ones.
[5] Data Labelling: Annotate pairs of questions as either duplicate or non-duplicate. You can utilize existing duplicate question links provided in the dataset or implement a labelling mechanism based on similarity thresholds. Ensure a balanced distribution of duplicate and non-duplicate question pairs in the labelled dataset.
[6] Data Splitting: Split the labelled dataset into training, validation, and testing sets. Ensure that each set contains a representative distribution of duplicate and non-duplicate question pairs.
[7] Model Selection and Training: Choose appropriate machine learning or deep learning models for duplicate question detection. Common models include Siamese networks, CNNs, RNNs, and Transformer-based architectures like BERT. Train the selected models using the training dataset. Fine-tune hyperparameters using the validation set to optimize model performance.
[8] Model Evaluation: Evaluate the trained models using appropriate evaluation metrics such as accuracy, precision, recall, F1-score, and ROC-AUC. Use the testing set to assess the generalization ability of the models. Compare the performance of different models and feature representations.
[9] Model Deployment: Develop an application or service that integrates the trained model to detect duplicate questions in real-time. Utilize the Stack Exchange API to fetch new questions and provide duplicate suggestions to users. Ensure scalability, reliability, and efficiency of the deployed system.
[10] Model Monitoring and Maintenance: Monitor the performance of the deployed model in production. Collect feedback from users and moderators to identify issues and areas for

improvement. Periodically retrain the model with new data to adapt to changing trends and user behaviour.

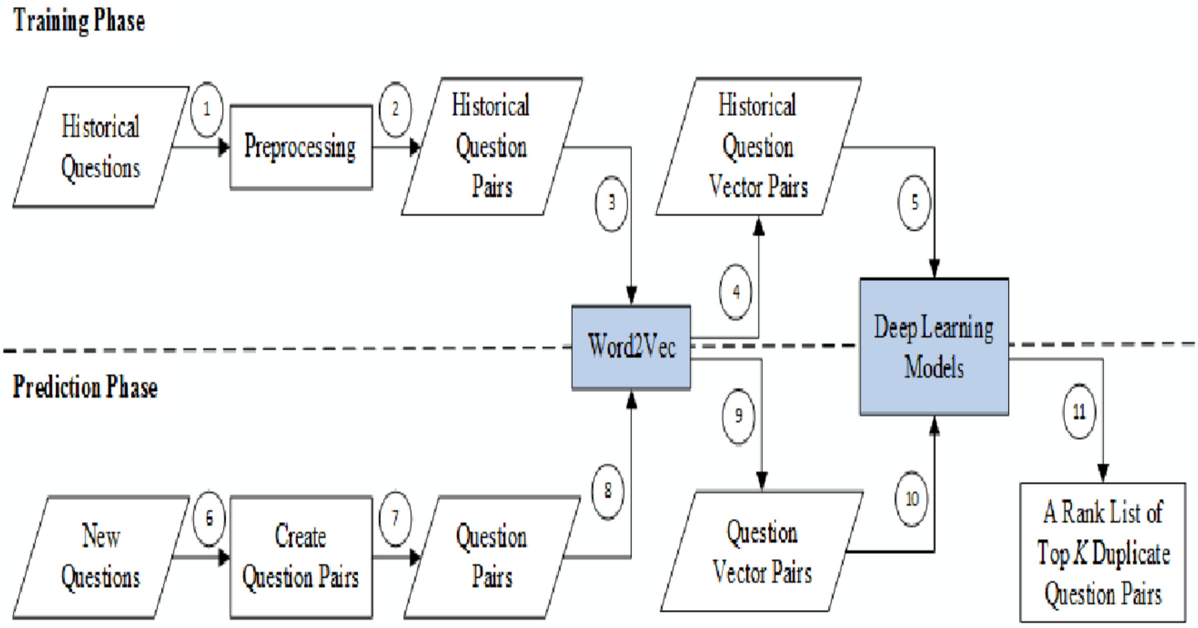


Fig (2) Methodology workflow

7. Algorithms Used:

Latent Dirichlet Allocation (LDA) is a generative probabilistic model commonly used for topic modelling, but it can also be adapted for text similarity tasks such as duplicate question detection. Below is a methodology for utilizing LDA algorithm in your duplicate question detection project:

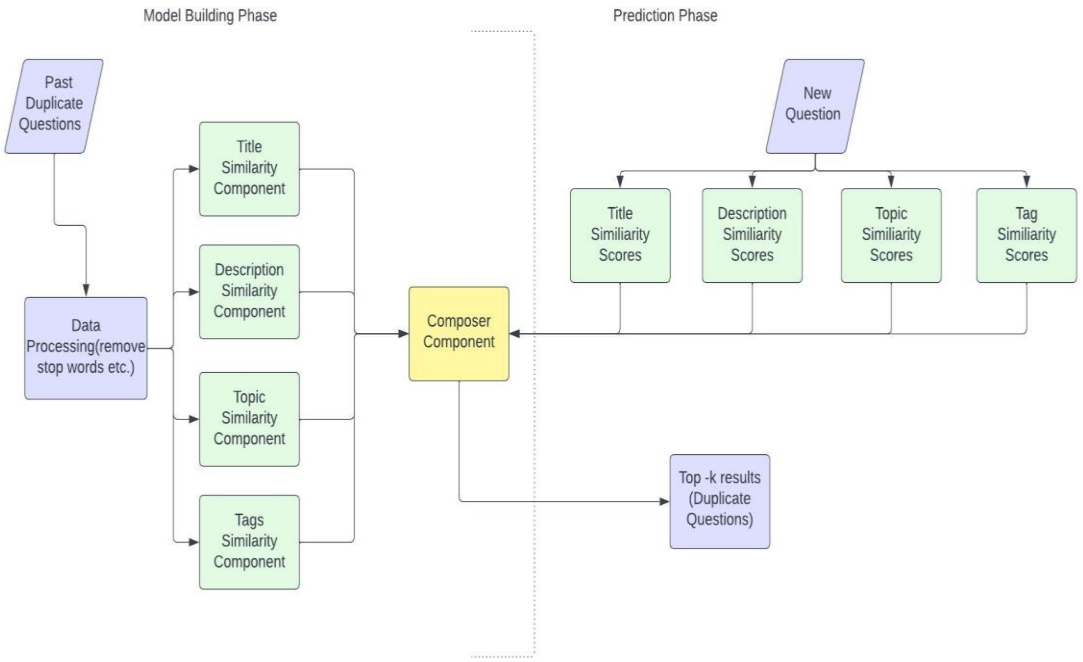

Model Building Phase
Prediction Phase
Past Duplicate Questions
Data Processing(remove stop words etc.)
Title Similarity Component
Description Similarity Component
Topic Similarity Component
Tags Similarity Component
Composer Component
New Question
Title Similiarity Scores
Description Similiarity Scores
Topic Similiarity Scores
Tag Similiarity Scores
Top -k results (Duplicate Questions)

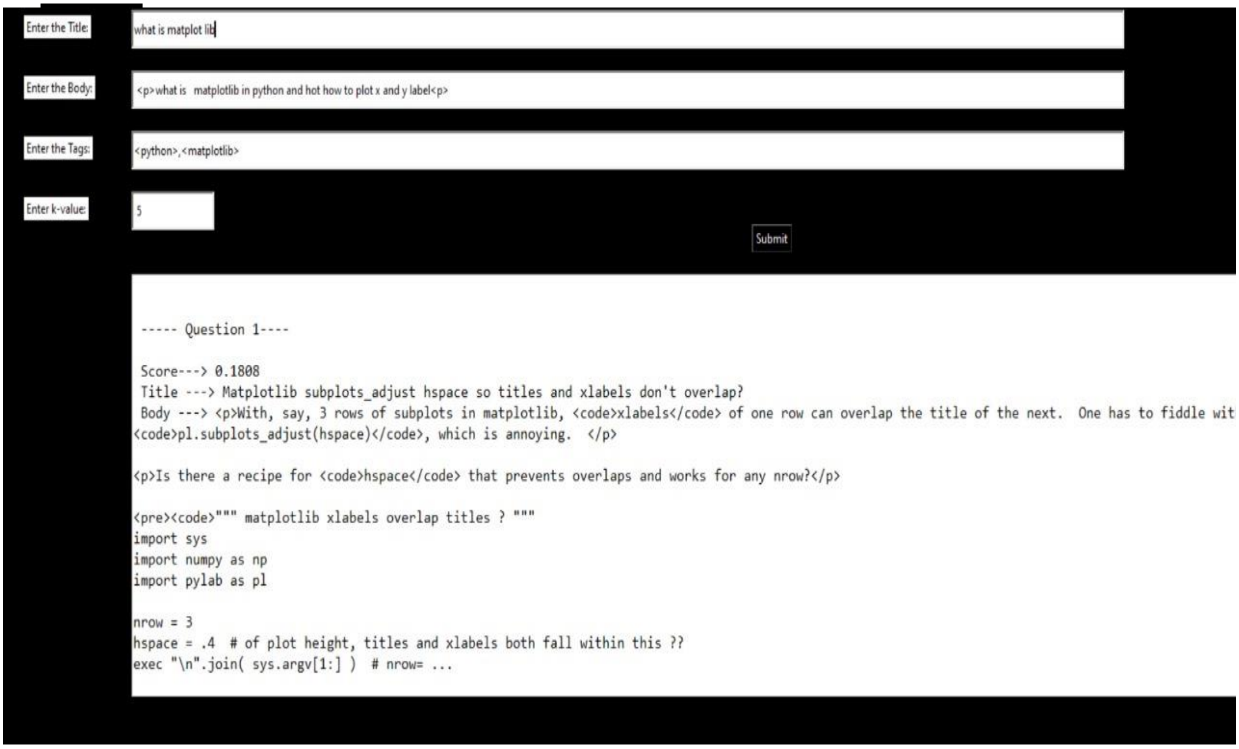
Fig (3) LDA Algorithm workflow

[1] Preprocessing and Data Preparation: Clean and preprocess the text data from the Stack Exchange dataset. Tokenize the text into words or phrases, remove stop words, and perform stemming or lemmatization. Prepare the data in a format suitable for LDA, typically a document-term matrix where rows represent documents (questions) and columns represent terms (words).
[2] Model Training: Train an LDA model on the pre-processed dataset. Set the number of topics as a hyperparameter. This may require experimentation to determine the optimal number of topics. LDA infers the topic distribution for each document and the word distribution for each topic based on the observed data.
[3] Topic Assignment: For each question in the dataset, infer its topic distribution using the trained LDA model. Assign the question to the topic with the highest probability.
[4] Similarity Calculation: Calculate the similarity between pairs of questions based on their topic distributions. One common approach is to use cosine similarity or other distance metrics to compare the topic distributions of question pairs.
[5] Thresholding and Decision Making: Set a similarity threshold to determine whether two questions are duplicates. Questions with similarity scores above the threshold are considered potential duplicates, while those below the threshold are not. You may adjust the threshold based on the desired balance between precision and recall.

8. Conclusion:

In conclusion, our project aimed to develop a user-friendly interface for Stack Overflow Duplicate Question Detection. We designed a Tkinter-based GUI allowing users to input a title, body, and tags, and specify the value of k. The workflow involved. Input Collection is used to provide question details. Processing Title, body, and tags undergo preprocessing. Retrieve top-k similar questions based on input. Results are displayed in the output section. The interface provides a seamless experience for users, enhancing efficiency in identifying duplicate questions on Stack Overflow.

9. Result:



Enter the Title: what is matplot lib

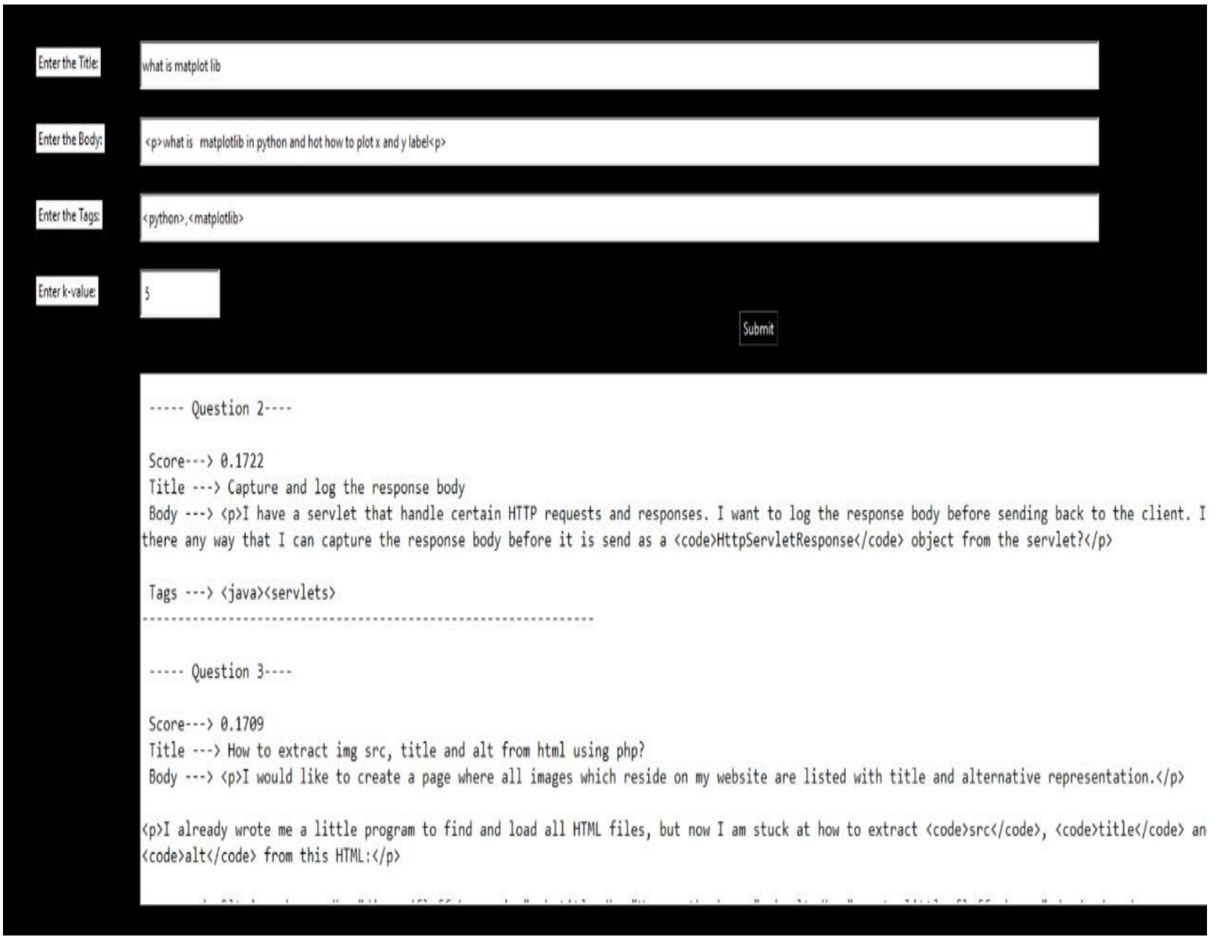
Enter the Body: <p>what is matplotlib in python and hot how to plot x and y label<p>

Enter the Tags: <python>,<matplotlib>

Enter k-value: 5

```
 ----- Question 1----

 Score---> 0.1808
 Title ---> Matplotlib subplots_adjust hspace so titles and xlabels don't overlap?
 Body ---> <p>With, say, 3 rows of subplots in matplotlib, <code>xlabels</code> of one row can overlap the title of the next.  One has to fiddle wit
<code>pl.subplots_adjust(hspace)</code>, which is annoying.  </p>

<p>Is there a recipe for <code>hspace</code> that prevents overlaps and works for any nrow?</p>

<pre><code>""" matplotlib xlabels overlap titles ? """
import sys
import numpy as np
import pylab as pl

nrow = 3
hspace = .4  # of plot height, titles and xlabels both fall within this ??
exec "\n".join( sys.argv[1:] )  # nrow= ...
```

Fig (4)



Enter the Title: what is matplot lib

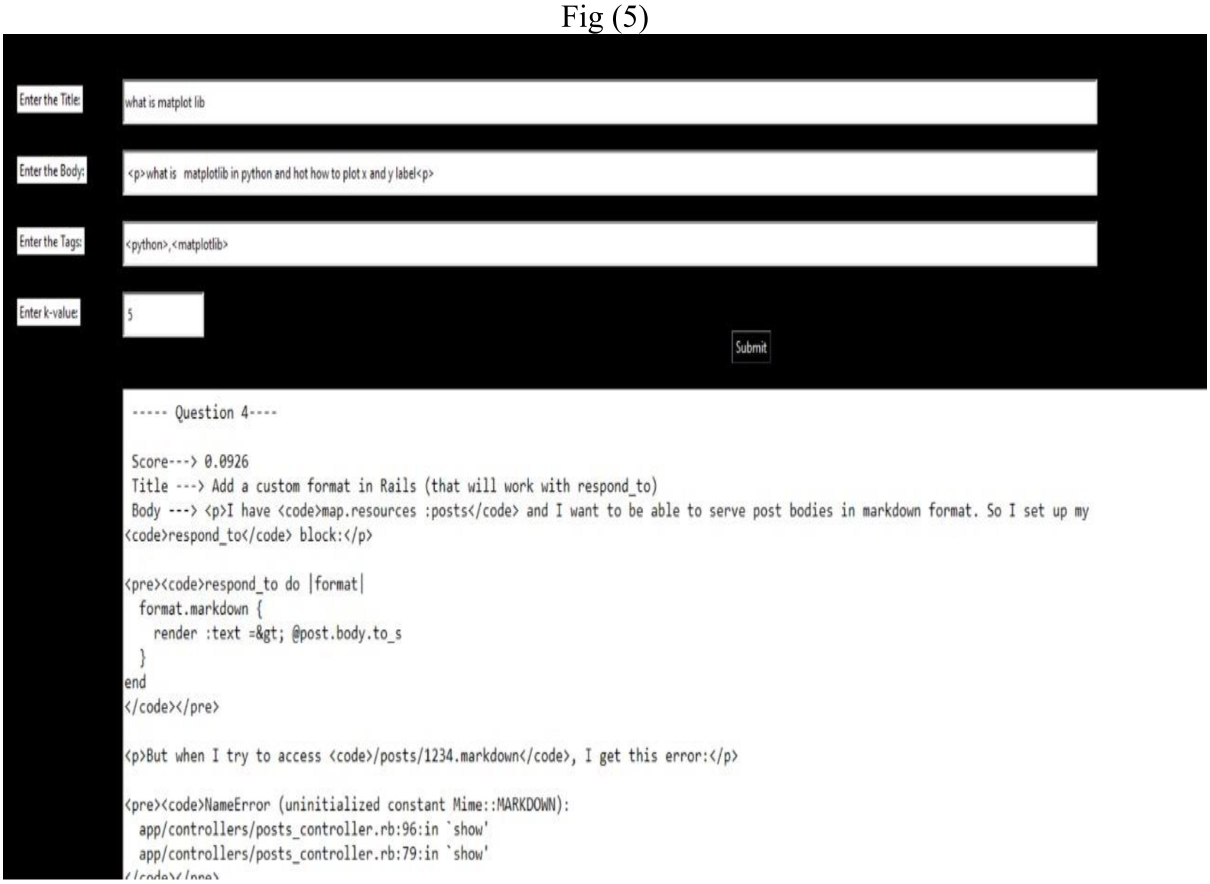
Enter the Body: <p>what is matplotlib in python and hot how to plot x and y label<p>

Enter the Tags: <python>,<matplotlib>

Enter k-value: 5

```
----- Question 2----

 Score---> 0.1722
 Title ---> Capture and log the response body
 Body ---> <p>I have a servlet that handle certain HTTP requests and responses. I want to log the response body before sending back to the client. I
there any way that I can capture the response body before it is send as a <code>HttpServletResponse</code> object from the servlet?</p>

 Tags ---> <java><servlets>
--------------------------------------------------------------

 ----- Question 3----

 Score---> 0.1709
 Title ---> How to extract img src, title and alt from html using php?
 Body ---> <p>I would like to create a page where all images which reside on my website are listed with title and alternative representation.</p>

<p>I already wrote me a little program to find and load all HTML files, but now I am stuck at how to extract <code>src</code>, <code>title</code> an
<code>alt</code> from this HTML:</p>
```

Fig (5)



Enter the Title: what is matplot lib

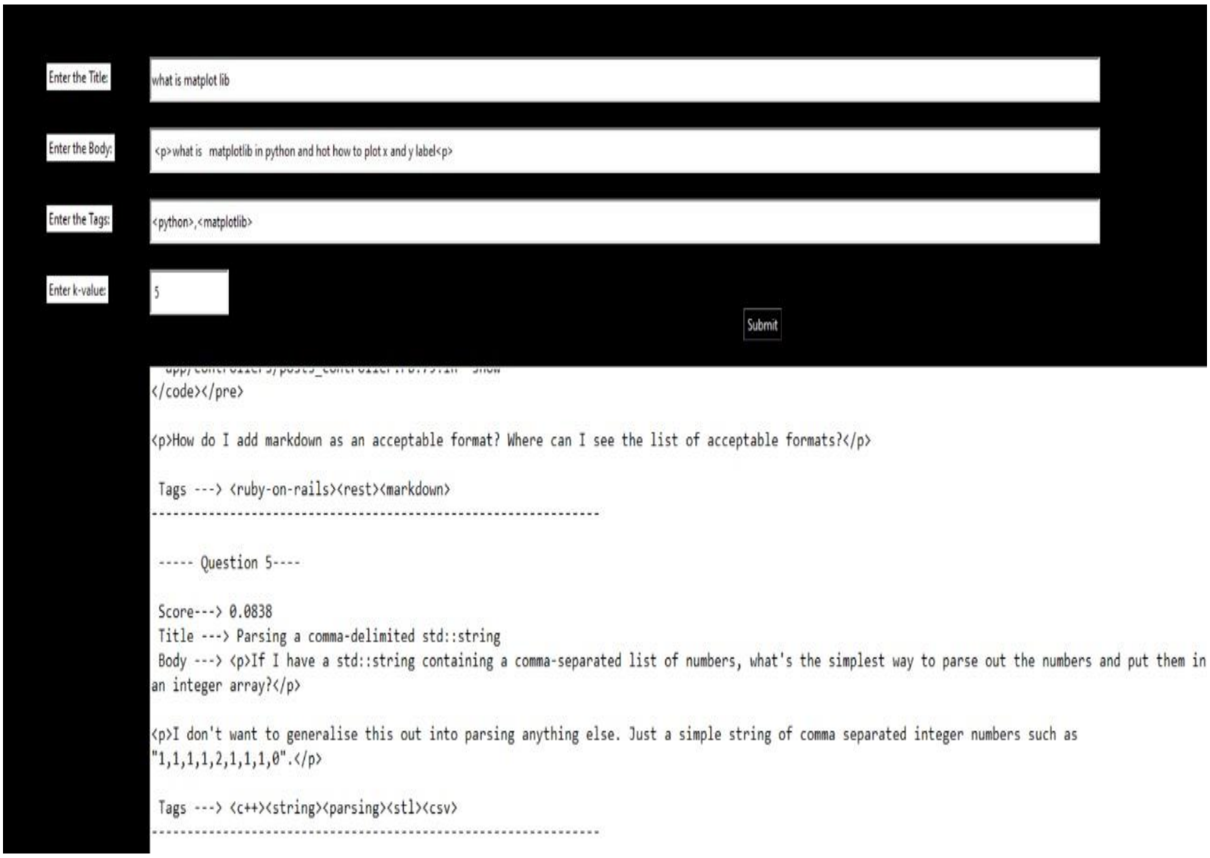
Enter the Body: <p>what is matplotlib in python and hot how to plot x and y label<p>

Enter the Tags: <python>,<matplotlib>

Enter k-value: 5

```
 ----- Question 4----

 Score---> 0.0926
 Title ---> Add a custom format in Rails (that will work with respond_to)
 Body ---> <p>I have <code>map.resources :posts</code> and I want to be able to serve post bodies in markdown format. So I set up my
<code>respond_to</code> block:</p>

<pre><code>respond_to do |format|
  format.markdown {
    render :text =&gt; @post.body.to_s
  }
end
</code></pre>

<p>But when I try to access <code>/posts/1234.markdown</code>, I get this error:</p>

<pre><code>NameError (uninitialized constant Mime::MARKDOWN):
  app/controllers/posts_controller.rb:96:in `show'
  app/controllers/posts_controller.rb:79:in `show'
</code></pre>
```

Fig (6)



Enter the Title: what is matplot lib

Enter the Body: <p>what is matplotlib in python and hot how to plot x and y label<p>

Enter the Tags: <python>,<matplotlib>

Enter k-value: 5

```
</code></pre>

<p>How do I add markdown as an acceptable format? Where can I see the list of acceptable formats?</p>

 Tags ---> <ruby-on-rails><rest><markdown>
-------------------------------------------------------------


 ----- Question 5----

 Score---> 0.0838
 Title ---> Parsing a comma-delimited std::string
 Body ---> <p>If I have a std::string containing a comma-separated list of numbers, what's the simplest way to parse out the numbers and put them in
an integer array?</p>

<p>I don't want to generalise this out into parsing anything else. Just a simple string of comma separated integer numbers such as
"1,1,1,1,2,1,1,1,0".</p>

 Tags ---> <c++><string><parsing><stl><csv>
-------------------------------------------------------------
```

Fig (7)

10. <u>Future Scope</u>:

[1] Enhanced Topic Modelling Techniques: Explore advanced topic modelling techniques beyond LDA, such as the Hierarchical Dirichlet Process (HDP) or Dynamic Topic Models (DTM). These methods may offer better modelling of complex relationships and temporal dynamics within the question corpus.

[2] Incorporating Contextual Information: Integrate additional contextual information into the model, such as user profiles, question tags, or temporal information. This could help improve the accuracy of duplicate question detection by considering the context in which questions are asked and answered.

[3] Semi-Supervised Learning: Investigate semi-supervised learning approaches to leverage both labelled and unlabelled data for training the model. Techniques such as self-training or co-training could help improve model performance by exploiting the abundance of unlabelled data available in the Stack Exchange dataset.

11. <u>Reference</u>:

[1] D. Correa and A. Sureka, Chaff from the wheat: Characterization and modelling of deleted questions on stack overflow, in Proc. 23rd Int. Conf. World Wide Web (WWW), Jan. 2014, pp. 631–642.

[2] Y. Zhang, D. Lo, X. Xia, and J.-L. Sun, Multi-factor duplicate question detection in stack overflow, J. Computer. Sci. Technol., vol. 30, no. 5, pp. 981–997, Sep. 2015.

[3] DM Blei, AY Ng, MI Jordan, Journal of machine Learning research, 2013. We describe latent Dirichlet allocation (LDA), a generative probabilistic model for collections of discrete data such as text corpora. LDA is a three-level hierarchical Bayesian model, in which each item of a collection is modelled as a finite mixture over an underlying set of topics.

[4] Naomi S. Altman. 1992. An introduction to kernel and nearest-neighbour nonparametric regression. The American Statistician 46, 3 (1992), 175--185.

[5] Yue Liu, Ju Yang, Mingjun Liu, Recognition of QR Code with mobile phones, Control and Decision Conference, CCDC 2008. Chinese, pp. 203 - 206, 2-4 July 2008.

[6] Muhammad Ahasanuzzaman, Muhammad Asaduzzaman, Chanchal K. Roy, and Kevin A. Schneider. Mining duplicate questions in stack overflow. In Proceedings of of the MSR 2016. ACM, Austin, Texas, USA, 402--412.

[7] Gianni Amati and Cornelis Joost Van Rijsbergen. 2002. Probabilistic models of information retrieval based on measuring the divergence from randomness. ACM Transactions on Information Systems 20, 4 (2002), 357--389.

[8] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on freebase from question-answer pairs. In Proceedings of the EMNLP 2013. ACL, Seattle, Washington, USA, 1533--1544.

[9] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. Journal of Machine Learning Research 3 (2003), 993--1022.

[10] Xin Cao, Gao Cong, Bin Cui, Christian S. Jensen, and Quan Yuan. 2012. Approaches to exploring category information for question retrieval in community question-answer archives. ACM Transactions on Information Systems 30, 2 (2012), 7.

[11] Tony F. Chan, Gene Howard Golub, and Randall J. LeVeque. Updating formulae and a pairwise algorithm for computing sample variances. In Proceedings of the COMPSTAT 1982. Springer, Physical, Heidelberg, 30--41.

[12] Stéphane Clinchant and Éric Gaussier. Information-based models for ad hoc IR. In Proceedings of the SIGIR 2010. ACM, Geneva, Switzerland, 234--241.

[13] Fred Jelinek and Robert L. Mercer. Interpolated estimation of Markov source parameters from sparse data. In Proceedings of the PRNI 1980. North Holland, Amsterdam, Netherlands, 381--397.

[14] Nitin Madnani, Joel R. Tetreault, and Martin Chodorow. Re-examining machine translation metrics for paraphrase identification. In Proceedings of the NAACL 2012. ACL, Montréal, Canada, 182--190.