Unsupervised Question Duplicate and Related Questions Detection in e-learning platforms

Maksimjeet Chowdhary*
Sanyam Goyal*
maksimjeet20566@iiitd.ac.in
sanyam20116@iiitd.ac.in
Indraprastha Institute of Information Technology
Delhi, India

Venktesh V
venkteshv@iiitd.ac.in
Indraprastha Institute of Information Technology
Delhi, India

Mukesh Mohania
mukesh@iiitd.ac.in
Indraprastha Institute of Information Technology
Delhi, India

Vikram Goyal
vikram@iiitd.ac.in
Indraprastha Institute of Information Technology
Delhi, India

ABSTRACT

Online learning platforms provide diverse questions to gauge the learners' understanding of different concepts. The repository of questions has to be constantly updated to ensure a diverse pool of questions to conduct assessments for learners. However, it is impossible for the academician to manually skim through the large repository of questions to check for duplicates when onboarding new questions from external sources. Hence, we propose a tool *QDup* in this paper that can surface near-duplicate and semantically related questions without any supervised data. The proposed tool follows an unsupervised hybrid pipeline of statistical and neural approaches for incorporating different nuances in similarity for the task of question duplicate detection. We demonstrate that *QDup* can detect near-duplicate questions and also suggest related questions for practice with remarkable accuracy and speed from a large repository of questions. The demo video of the tool can be found at https://www.youtube.com/watch?v=loh0_-7XLW4.

CCS CONCEPTS

• **Information systems** → *Information retrieval*; • **Applied computing** → *Document searching*.

KEYWORDS

semantic similarity, duplicate detection

ACM Reference Format:
Maksimjeet Chowdhary, Sanyam Goyal, Venktesh V, Mukesh Mohania, and Vikram Goyal. 2018. Unsupervised Question Duplicate and Related Questions Detection in e-learning platforms. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX).* ACM, New York, NY, USA, 4 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

The e-learning platforms usually curate a large repository of questions across subjects, chapters, and topics for conducting assessments to test the understanding of the learner. These repositories are constantly augmented with new questions. The new questions could be collected in batches from other platforms or external sources. They could also be added manually by the academicians. When new questions are added, there are cases of them being near-duplicates or related to existing questions in the data repository. It is impossible for the academicians to manually skim through the entire repository to check for duplicates. Hence, in this work, we propose a tool with support for bulk on-boarding of questions while surfacing duplicate questions already present in the database.

The duplicate question detection task, particularly in the context of e-learning platforms, is a significant challenge due to the nature of the questions. Two questions can differ in entities or technical concepts though their verbiage and the rest of the semantics could be similar. In certain cases, though the questions are centered around the same entity and have mostly similar verbiage, the answers could be different. For example, the questions *What is GDP?* and *What is the significance of GDP?* might have high Jaccard or cosine similarity but are not duplicate questions. Hence, to encompass the mentioned scenarios, we define two questions to be duplicates of each other if they satisfy all of the following conditions:

- The questions are lexically similar and have synonymous keyphrases or entities.
- The questions are semantically related.
- The correct answers to both the questions are equivalent

We also recommend related questions to aid the academicians in generating diverse questions for assessments. For instance, the questions *What is the strongest bone in the body?* and *What is the weakest bone in the body?* are related questions.

The duplicate text detection [2–4, 7, 10] is a well explored problem. These approaches range from comparing topics obtained through topic modelling [10], comparing syntactic structure [4] to neural

*Both authors contributed equally to this research.

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06...$15.00
https://doi.org/XXXXXXX.XXXXXXX

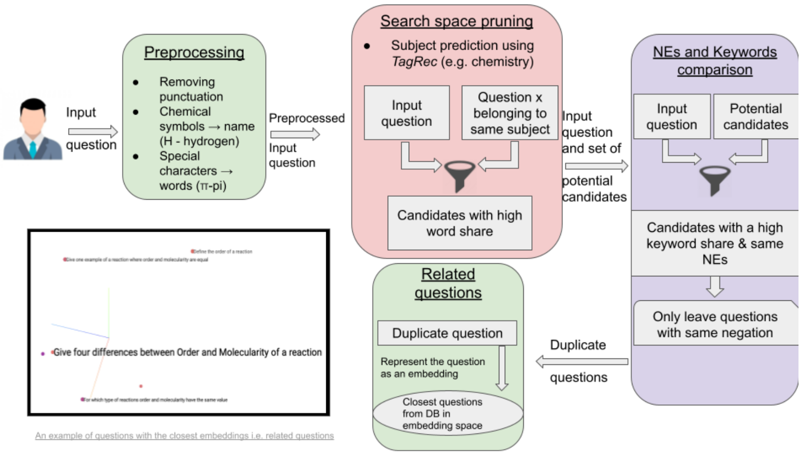

Input
question
Preprocessing
Removing punctuation
Chemical symbols → name (H - hydrogen)
Special characters → words (π-pi)
Preprocessed
Input question
Search space pruning
Subject prediction using TagRec (e.g. chemistry)
Input question
Question x belonging to same subject
Candidates with high word share
Input question and set of potential candidates
NEs and Keywords comparison
Input question
Potential candidates
Candidates with a high keyword share & same NEs
Only leave questions with same negation
Duplicate
questions
Related questions
Duplicate question
Represent the question as an embedding
Closest questions from DB in embedding space
Define the order of a reaction
Give one example of a reaction where order and molecularity are equal
Give four differences between Order and Molecularity of a reaction
For which type of reactions order and molecularity have the same value
An example of questions with the closest embeddings i.e. related questions

Figure 1: Duplicate Question Detection Pipleine

IR based methods [3]. However, these approaches consider only uni-dimensional aspects of similarity, as mentioned earlier, and fail to identify duplicates in other scenarios where the questions only differ in entities. They also require significant amounts of the labeled dataset where questions are labeled as duplicates like CQADupStack [6, 8], which is not available in the problem setting explained in this paper.

The pipeline proposed in this paper is unsupervised and efficient in that it does not require any training. Since our approach is hybrid and uses a combination of classical and neural IR approaches, it is also efficient at inference time. The overview of the proposed pipeline can be seen in Figure 1. In summary, our core contributions are:

- We propose an unsupervised approach for near-duplicate detection in online learning platforms to enable smooth onboarding of new questions. We also recommend related questions for serving diverse questions.
- We develop and release an easy-to-use tool that can support both individual and bulk on-boarding of questions.

2 SYSTEM DESIGN

In this section, we describe the methodology used for searching for duplicate questions with respect to a large existing question repository. Given an input question $q_{new} = \{x_1, x_2 ... x_n\}$ of sequence length n our goal is to surface exact duplicate questions $qdup_{exact}$, near-duplicates $qdup$ and related questions q_{rel}. We present an unsupervised pipeline that uses an iterative elimination approach, removing questions that are certainly non-duplicates and retaining exact or near-duplicate questions. The proposed approach is different from existing paraphrase identification or duplicate detection approaches as it covers different aspects of similarity in a single pipeline with no supervised data. The pipeline proposed is shown in Figure 1. The pipeline consists of the following stages :

(1) Preprocessing and hierarchical learning taxonomy tagging
(2) Jaccard similarity between questions tagged with similar learning taxonomy.
(3) Named Entity Recognition for computing entity differences.
(4) Overlap of key concepts obtained through concept extraction algorithm and negation detection.

2.1 Preprocessing and Indexing by Hierarchical Learning Taxonomy

Given a question q_{new} as input, we preprocess the question, such as sentence level tokenization, removing HTML tags, and non-alphanumeric characters, and removing punctuation marks. The database includes questions asked in high school and belongs to various subjects, including chemistry, physics etc.

Therefore we normalize chemical element abbreviations and symbols to their complete form (Cl ⟶ chlorine, pi ⟶ π etc.) using a dictionary $dict_{sym}$ to ensure consistency resulting in q_{norm}.

$$q_{norm} = f_{norm}(q_{new})$$

$$S \longleftarrow tokenize(q_{new})$$

$$f_{norm} = dict_{sym}[s_i] \; for \; s_i \; in \; S$$

After preprocessing the input question, we tag the input question to its standardized hierarchical learning taxonomy of form subject - chapter - topic using the TagRec [9] model. The TagRec approach follows a two-tower transformers-based architecture that aligns the vector subspaces of the input question and the hierarchical learning taxonomy using a contrastive learning approach. We use this trained model to tag our database of questions and q_{new} in a zero-shot setting and index the questions according to the tags.

We extract the subject portion of the taxonomy to which the question belongs and query the complete database to return the candidate set $S_{cand} = \{ q_1 , q_2 , ... q_n \}$ of all the questions in the database that belong to the same subject. Our dataset primarily consists of questions from the subjects: Physics, Chemistry, Social Science, etc. For example, the question *How many π bonds are present in ferrocene?* belongs to the subject Chemistry.

2.2 Token level comparison

After getting the set S_{cand} of the questions belonging to the same subject as from the same hierarchy as the input question q_{new}, the model iterates over S_{cand} and checks for the $JaccardSimilarity$ measure.

More formally, Let q_1 , q_2 be two lists of tokens for the input questions, then Jaccard similarity between these two questions can be calculated as

$$J(q_1, q_2) = \frac{\#(q_1 \cap q_2)}{\#(q_1 \cup q_2)} \tag{1}$$

If the Jaccard similarity ($J(q_{new}, q_i)$) between the input question and a question from S_{cand} is less than a certain threshold ($J(q_{new}, q_i) < 0.4$) we remove that question from our search space S_{cand}. The threshold value of 0.4 was chosen after multiple iterations and validation of the results for the dataset that we worked on.

$$S_{cand} \leftarrow S_{cand} - q_i \ (if \ J(q_{new}, q_i) < 0.4)$$

If the Jaccard similarity is 1 we directly add that question to our exact duplicate question set ($qdup_{exact}$).

2.3 NER and comparison

To further partition S_{cand}, we remove questions with different named entities than those in q_{new}. For extracting the set of named entities (NE_q) of q_{new} we use spaCy, an implementation in Python.

$$NE_q = NER(q_{new})$$

Examples : Who is the CEO of Google ? → 'Google': ORG , Who is the CEO of Apple ? → 'Apple' : ORG

The Named Entity Recognition step performs a sequence labeling task where the noun phrases are tagged with 'PERSON', 'ORG', 'LOC', etc as applicable. Once extracted, the set of entities for q_{new} is compared to the set of entities NE_i for question q_i, where i = 1... $|S_{cand}|$. All those questions which have a non-empty difference set between NE_q and NE_i are removed from the search space (S_{cand}).

$$S_{cand} \leftarrow S_{cand} - q_i \ if \ NE_q \cap NE_i \neq \emptyset$$

2.4 Keyphrase extraction and calculating the overlap

Following the previous stages of the pipeline, the set S_{cand} = { q_1 , q_2 , ... q_n } is left of potential candidates for a duplicate question. The next stage of the pipeline (as shown in Figure 1) is to run an unsupervised method to automatically extract concept terms (keyphrases) from the input question q_{new} into a set KW_i. We leverage the EmbedRank algorithm [1] for extracting keyphrases. The proposed approach first extracts candidate phrases using POS tags and projects them and the original question q_{new} to a continuous vector space. It then computes the semantic relatedness between the question and the phrase representations and retrieves the top k keyphrases. For all the questions, we pre-compute the keyphrases and index them. We run a comparison to determine the percentage of keyphrases shared between KW_i, and the set of keyphrases extracted for each of the questions in set S_{cand}. Questions that have keyphrases sharing score of less than 0.7 (chosen after multiple validations) are eliminated from S_{cand}.

$$KW_i \leftarrow EmbedRank(q_{new})$$

$$S_{cand} \leftarrow S_{cand} - q_i \ if \ KW_{share} < 0.7$$

2.5 Negation detection

As a result of the previous steps, the set S_{cand} is much smaller in size and has questions very similar to q_{new}. However, we observed that multiple questions with similar verbiage exist though they differ by a negation resulting them having different answers. For example: *What is an example of a metal ?* and *What is not an example of a metal ?*

Similar cases might still be left in S_{cand}, and hence we check for the difference in negation. We compare q_{new} against each question in S_{cand} and eliminate any questions that may be the negation of q_{new} by ensuring that standard negation constructions, if any, are present in both samples being compared. This ensures that questions with high levels of Jaccard similarity and overlapping keyphrases shares but differing by a single negation are not identified as duplicates (false positives). After this stage, we assign the remaining questions to be duplicates.

2.6 Related Questions

The above-mentioned pipeline focuses on a higher recall by sacrificing precision since the problem statement focuses on e-learning platforms being able to rid their database of duplicates. An additional property of our tool is that these platforms can test students on their knowledge of the topic by retrieving related questions which center around the same or similar topics.

Such questions are referred to as *related questions* in this paper and are computed by utilizing the architecture of $all{-}mpnet{-}base{-}v2$ sentence transformers model[1]. In the approach, for a duplicate question q_{dup} of an input question q_{new}, we find the questions that have embeddings closest to $qdup_{exact}$ or q_{dup} (pre-computed in the database), measured by the cosine similarity between the two embedding vectors and return the 3 closest neighbors.

The results demonstrated that nearest neighbors search over a large database gave slow performance, which led us to leverage *ScaNN* (an efficient searching technique developed by [5]. The set of embeddings for all the questions in the database is precalculated (using the same $all - mpnet - base - v2$ model) and stored locally for higher efficiency during running.

For every input q_{new}, we have $qdup_{exact}$, $qdup$ and related questions $qdup_{rel}$ (q_{rel} is non-empty only if $qdup$ is non-empty).

3 DEMONSTRATION

We demo our tool from the perspective of it's ability to perform **near-duplicate detection**, analysis to gauge **usability** of the tool.

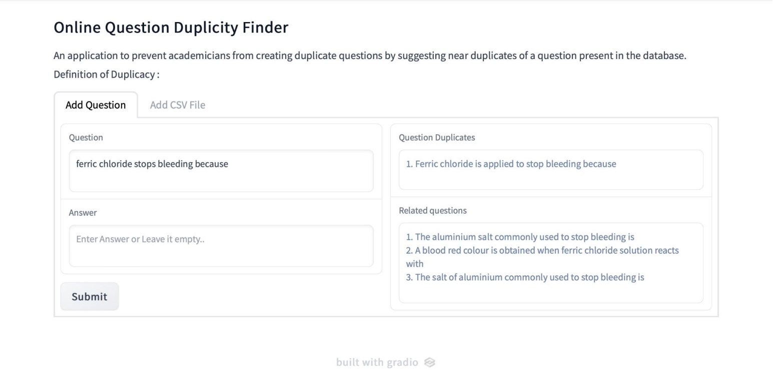
3.1 Dataset

The dataset we used consists of 114804 secondary high school questions from the CBSE (India) curated with the help of a leading e-learning platform. The dataset statistics are shown in Figure 3. It consists of questions from *science, social science, computer science, chemistry, physics and political science.*

3.2 Evaluation

The tool was evaluated on a set of 100 input questions by two independent researchers. The tool was provided with 100 random

[1]



Online Question Duplicity Finder

An application to prevent academicians from creating duplicate questions by suggesting near duplicates of a question present in the database.

Definition of Duplicacy :

Question: ferric chloride stops bleeding because

Answer: Enter Answer or Leave it empty..

Question Duplicates:

1. Ferric chloride is applied to stop bleeding because

Related questions:

1. The aluminium salt commonly used to stop bleeding is
2. A blood red colour is obtained when ferric chloride solution reacts with
3. The salt of aluminium commonly used to stop bleeding is

Figure 2: Screenshot of the tool *QDup*

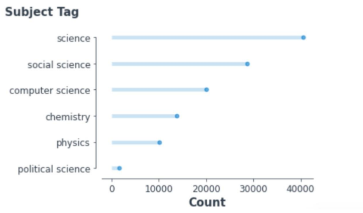


Figure 3: Dataset Statistics

Method	Accuracy (%)
QDup	**81.5**
keyphrases based	76.5
Closest neighbours	51.5

Table 1: Performance Evaluation for Duplicate Detection

questions across domains and the researchers were requested to label the correct duplicates as 1 or 0 in all other conditions. Similarly, the outputs from other approaches were also provided to the researchers for labeling. These approaches included nearest neighbor search for embeddings extracted with all-mpnet-base-v2 sentence embeddings model and comparison of keyphrases extracted using EmbedRank. We observed a Cohen’s kappa of **0.60**, **0.72** and **0.65** in the three scenarios, respectively indicating substantial agreement between annotators. We report the accuracy in Table 1. We observe that the proposed approach *QDup* outperforms classical keyphrases only or vector based nearest neighbor search methods.

3.3 Tool Ease of Use

We also conducted a user study with 14 well trained academicians. A screenshot of the tool is shown in Figure 2. We asked the users to rate the tool on a scale of 1-3 (lowest to highest) from aspects of *intuitiveness*, *responsiveness* and *relevance of output*. The *intuitiveness* metric indicates how intuitive and easy to use the interface is without external help. The *responsiveness* measures the response time and *relevance* measures how much the users think the output for the given questions are accurate duplicates. We observed that the average *intuitiveness* score is **2.46** and average *responsiveness* score is **2.78**. The average *relevance* score is **2.68**. We observe that the majority of the users find the tool easy to use.

4 CONCLUSION AND FUTURE WORK

In this paper, we propose a tool to find duplicates and related questions in a large repository. The proposed approach is resource and time efficient, and the interface is easy to use. In the future, we plan to use the data collected from this tool as weakly supervised data to train a bi-encoder transformer-based model in a contrastive setting to identify duplicate and related questions in one stage. We also plan to explore knowledge distillation and quantization approaches for efficient deployment of the model.

REFERENCES

[1] Kamil Bennani-Smires, Claudiu Musat, Andreea Hossmann, Michael Baeriswyl, and Martin Jaggi. 2018. Simple Unsupervised Keyphrase Extraction using Sentence Embeddings.

[2] Giovanni Da San Martino, Salvatore Romeo, Alberto Barroón-Cedeño, Shafiq Joty, Lluís Maàrquez, Alessandro Moschitti, and Preslav Nakov. 2017. Cross-Language Question Re-Ranking *(SIGIR ’17)*.

[3] Arpita Das, Harish Yenala, Manoj Chinnakotla, and Manish Shrivastava. 2016. Together we stand: Siamese Networks for Similar Question Retrieval. In *ACL*. Berlin, Germany, 378–387.

[4] Simone Filice and Alessandro Moschitti. 2018. Learning pairwise patterns in Community Question Answering. *Intelligenza Artificiale* 12 (2018), 49–65.

[5] Ruiqi Guo, Philip Sun, Erik Lindgren, Quan Geng, David Simcha, Felix Chern, and Sanjiv Kumar. 2019. Accelerating Large-Scale Inference with Anisotropic Vector Quantization.

[6] Doris Hoogeveen, Karin M. Verspoor, and Timothy Baldwin. 2015. CQADupStack: A Benchmark Data Set for Community Question-Answering Research *(ADCS ’15)*.

[7] Di Liang, Fubao Zhang, Weidong Zhang, Qi Zhang, Jinlan Fu, Minlong Peng, Tao Gui, and Xuanjing Huang. 2019. Adaptive Multi-Attention Network Incorporating Answer Information for Duplicate Question Detection *(SIGIR’19)*. 95–104.

[8] Preslav Nakov, Lluís Màrquez, Alessandro Moschitti, Walid Magdy, Hamdy Mubarak, Abed Alhakim Freihat, Jim Glass, and Bilal Randeree. 2016. SemEval-2016 Task 3: Community Question Answering. ACL, San Diego, California.

[9] Venktesh V, Mukesh Mohania, and Vikram Goyal. 2021. TagRec: Automated Tagging of Questions with Hierarchical Learning Taxonomy.

[10] Kai Zhang, Wei Wu, Haocheng Wu, Zhoujun Li, and Ming Zhou. 2014. Question Retrieval with High Quality Answers in Community Question Answering *(CIKM ’14)*. Association for Computing Machinery, New York, NY, USA, 371–380.