QueRe : Retrieval of Question-Answer from StackOverflow

Sakshi Choudhary
sakshi20040@iiitd.ac.in

Shreya Garg
shreya20133@iiitd.ac.in

Soumam Banerjee
soumam20043@iiitd.ac.in

Harshita Sahu
harshita20087@iiitd.ac.in

1. ABSTRACT

Community based QA platforms like, ask-Ubuntu, Stack Overflow, Gate Overflow and likewise bestows us with plethora of knowledge and information, although mining of relevant questions depending on the query provided by user is a daunting task for such a large corpus of document collection,which makes it a tremendously popular research topic besides applying to production. The fundamental issue this problem statement encounters is to measure the semantic similarity between the query and the questions present in the document collection,which is previously solved/answered by other experienced users. Traditional methods like BOW(Bag-of-words) did not capture the dependencies as well as the synonyms between the query and questions. The proposed work suggests how we can vectorize a query/question of any length into same dimension and leverage the vectors to find out cosine-similarity between them and accordingly rank them. Hereby, we have also used the potential of TF-idf based approach to generate a final rank list based on our query. Experiments are conducted on 10% of stack-overflow data, which yielded us optimal results.

2. INTRODUCTION

In the recent past, QA platforms has immensely boomed so has their dataset of document collection. It being a structured data has become a wealthy resource in the domain of NLP and IR research. Decade ago the answers were short and concise in nature and so were the queries but with progress of time and advent of 2.0 , more complex questions were asked which traditional QA platforms fail to provide satisfactory answers. Large-scale QA archives thus now have dominated the field. These platforms ranges from customer support FAQ's of companies for particular products to websites like Reddit, Quora, Live QA, Stack Overflow,etc. These platforms are popularly referred as Community based Question answering services. Here, people gets the answer specific to the questions and not the list of relevant documents which makes life much simpler. To unleash the full potential of these platforms its is important that not only textually similar questions are fetched but also the semantically questions are retrieved . As most of the traditional models were based on Bag-of-words(BOWs) techniques and term weighting factors like tf-idf, BM-25 they were not able to capture the semanticity of the questions. Although we see most of the times the user generated queries are short and we need to uses it to its maximum potential. Traditional models don't provide sufficient word co-occurrence or context shared information for similarity scoring. due to which they are not as effective.

Recently, researchers have used transnational models to capture word , phrase semanticity , but being word specific many a times it misses out on the overall semanticity of the whole query , this is one of the reason why till date chatbots can't understand sarcasm well! Moreover these platforms are generally monolingual. We have focused our work to solve on these minute issues which tends to provide us tremendously better results than our baseline models which leveraged upon Tf-idf weight averaging of word based vectors.

The rest of the paper is organised as follows. Section 3 comprises of Literature Review. Section 4 is to depict the methodology proposed by us, while section 5 and 6 are for evaluation and conclusion respectively.

3. LITERATURE REVIEW

Over the years,the number of user's queries have increased with the advent of new technologies. Consequently, the Community Question Answering systems are becoming popular everyday. However, these systems should be equipped with powerful retrieval models to enhance the user's experience. Hence there is a need of efficient retrieval system.

Many researchers aimed to enhance the search relevance of CQA systems when the user's query is short. Studies indicate that short input query builds up a lexical gap between the user's query and past archives of questions. Chen et. al [2] addressed this issue and thereby proposed a combination of traditional language models along with a semantic-graph based topic model that outperformed state-of-the-art language models for question retrieval.However, this performed well on the General QA system(Yahoo Answers!), their approach has not been tested over domain specific QA systems.

The other challenges faced are mismatching due to shorter length questions and inefficiency of existing approaches to capture contextual information and semantic information between words. Nouha Outman Et al [7] proposed representation of questions in the form of word embeddings where word embeddings capture the semantic relation in the questions. These word embeddings of the question are weighted using TF-IDF and averaged. Cosine Similarity is applied to rank on the questions based on the weighted vector based word representations where higher cosine similarity score shows more semantic similarity between the queried question and a previous posted question from corpus.

A group of researchers [9] focused on solving the word verboseness in queries, while performing search on CQA platforms and also distinguish between the key concept from the non-key ones.There were two components, first component detect the key concept in query using ranking based method. The second component then automatically explores the key concept paraphrasing using pivot language translation approach from multiple language resources. They collected large question dataset from Yahoo! Answers which consist of 11,23,034 questions for question retrieval.
The key concept paraphrase based question retrieval model outperformed the state-of-the-art models in question retrieval task.

Some researchers [5] describe the work done for the SemEval 2017 Task 3 on Community Question Answering. It had four subtasks from which Subtask B was Question-Question Similarity. The English Dataset was taken from Qatar Living Forum which was further manually annotated, where all the examples were manually labeled by a community of annotators using crowdsourcing platform. In subtask B, for a given new question and set first 10 related questions from the forum retrieved by a search engine,the goal is to arrange these questions according to the similarity with respect to the original question. Mean-Average Precision(MAP) was taken as the evaluation metric.The winner of this task was team *SimBow*, they used logistic regression on a rich combination of different unsupervised textual similarity built using a relation matrix based on standard cosine similarity between bag-of-words and other semantic or lexical relation. They achieved a MAP of 47.22 and were winners of this task.

Recently, Google proposed Universal Sentence Encoder [1] for efficiently encoding sentences into high dimensional vector which can be used for text classification, semantic similarity, clustering etc.

In 2019, Sentence-BERT [8] outperformed the state-of-the-art sentence encoding methods.The Researchers claimed that SBERT magnificiently reduced the time taken to find the most similar pair of sentences from 65 hours with BERT / RoBERTa to about 5 seconds with SBERT, while maintaining the accuracy from BERT.

In 2018,the authors of [3] released a pretrained word2vec model that has been trained over 15GB of stackoverflow posts data. Recent research in Microsoft[4] led to the development of a benchmark data set that can be utilized for learning retrieval models for generation of similar question-answer pairs based on user's query. The authors demonstrated the strength of learning approach in data set building over the state-of-the-art non-learning approaches.

The agenda behind WE-COSIM [6] model, was to replicate each question in the dataset as a dense 200-dimensional vector. Although, traditional methods such as Bag Of Words (BOWs) were used but the proposed deep-learning based method not only considers the term/document frequency but also the vector representation of each word using which, we will generate a Bag of embedded words (BOEWs). In order to convert each and every word to its corresponding vector representation. They have used CBOW to generate word vectors and thereafter did TF-TDF based weighting to get a 200 dimensional vector for each question independently.Finally cosine similarity was used to retrieve similarity scores between a query with all the questions in the document collection and finally sorted to produce top-10 similar questions. The model produced 0.3333 for P@5 and 0.3 for P@10 on the test queries which is not at all impressive as per human standards of retrieval.

4. METHODOLGY

We propose "QueRe", an effective end-to-end Question-Answer retrieval system which will be capable of retrieving a ranked list of relevant Question-Answer pairs where the output will be ranked on the basis of relevant questions, overcoming the challenges of the existing systems. The proposed model presents a hybrid approach, combining the expediencies of keyword-based and semantic similarity based search.

The overall architecture of the proposed model is shown in Figure 4. Briefly, the workflow of the proposed system can be described in the following systems:

1. *Input:* The user query will be taken in the form of text. The input text can be multilingual. The input text is then converted to English language and then further preprocessing steps mentioned below are performed.

2. *Encoding for semantic search:* The preprocessed input obtained is passed to the Universal sentence encoder which transforms the query into sentence encoding. The collection of past archives of questions available in the corpus are also passed through the sentence encoder and transformed into sentence encodings as well. To find the list of relevant questions, cosine similarity score is calculated between sentence embedding of query and each archived question.

3. *Elastic Search for keyword based matching:* Elastic search is used for keyword based matching. The questions in the corpus are stored in the form of Inverted index such that on the receipt of user's query , hashing is performed for obtaining faster results which in turn provides us with the list of questions on the basis of keyword search.

4. *User Interface:* We built a web application using PywebIO as a prototype for our proposed model. It inputs a query (supports 36 languages) and generates a rank list based on descending order of their score. Besides, retrieving the questions, it also retrieves the answer having maximum score for that particular matched question from the corpus. The retrieved answers are downloadable. It has a feedback form ,as we are open to suggestions from our users. Our web-app was made public using ngrok which tunneled their server to our local host.

4.1 Dataset

The baseline model uses the "StackSample: 10% of Stack Overflow Q&A." dataset for demonstration of results.The dataset consists of three main components described as follows:

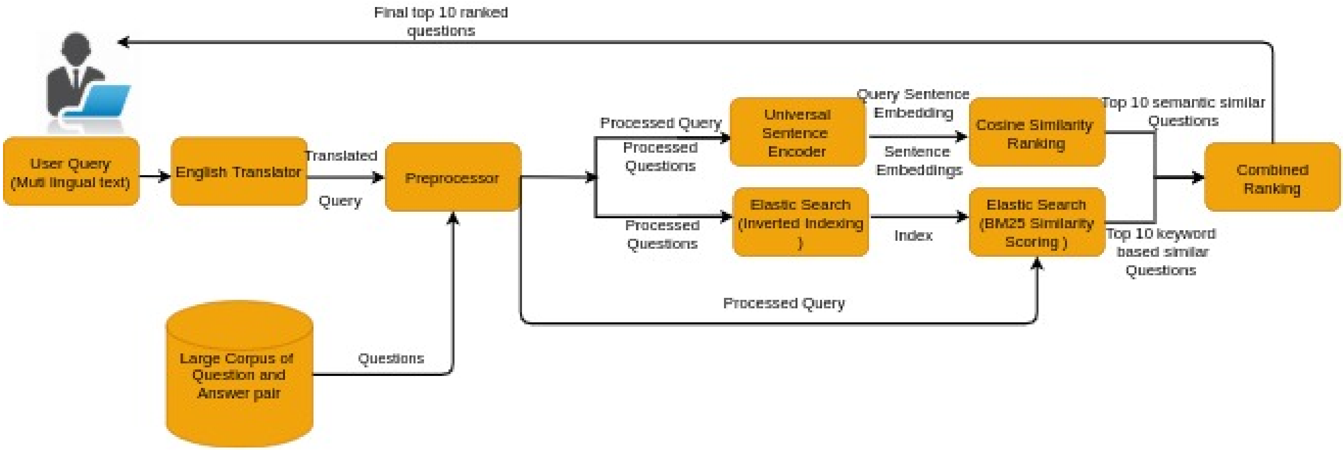

Final top 10 ranked questions
User Query (Muti lingual text)
English Translator
Translated
Query
Preprocessor
Processed Query
Processed Questions
Universal Sentence Encoder
Query Sentence Embedding
Sentence Embeddings
Cosine Similarity Ranking
Top 10 semantic similar Questions
Processed Questions
Elastic Search (Inverted Indexing)
Index
Elastic Search (BM25 Similarity Scoring)
Top 10 keyword based similar Questions
Combined Ranking
Processed Query
Large Corpus of Question and Answer pair
Questions

Figure 1: Proposed Model

(a) **Questions:** information regarding title, body, creation date, closed date (if applicable), score, and owner ID for all non-deleted Stack Overflow questions whose Id is a multiple of 10.

(b) **Answers:** contains the body, creation date, score, and owner ID for each of the answers to these questions. The ParentId column links back to the Questions table.

(c) **Tags:** contain the tags on each of these questions

4.2 Question Preprocessing

From the given dataset, we have taken question id, question title and its body for preprocessing.

(a) **Question title preprocessing:** Preprocessing steps involves conversion to lowercase, expanding the contractions, word tokenization, stop word removal.

(b) **Question body preprocessing:** Preprocessing steps involves conversion to lowercase, expanding the contractions, replacing newline with space, tag removal, removal of codeblocks, Url removal, word tokenization and stopward removal.

4.3 Best Answer Extraction

The aim is to return relevant question-answer pair, but there are multiple answer corresponding to single question. Thus only a single answer having highest score among all the other answers of the same question is paired and presented as output with the question.

4.4 Sentence Embedding learning using USE

A sentence embedding is a vector-representation providing semantic information of the whole sentence . We have used Univseral Sentence Encoder to obtain the sentence encoding. The sentence is firstly tokenized using the Penn Treebank(PTB) tokenizer. Then the encoder encodes these tokens into a 512-dimension dense sentence embedding. We have taken preprocessed question title and its body as a single sentence and generated sentence embedding. Similarly the preprocessed user query is also treated as a sentence.

4.5 Cosine Similarity based Ranking

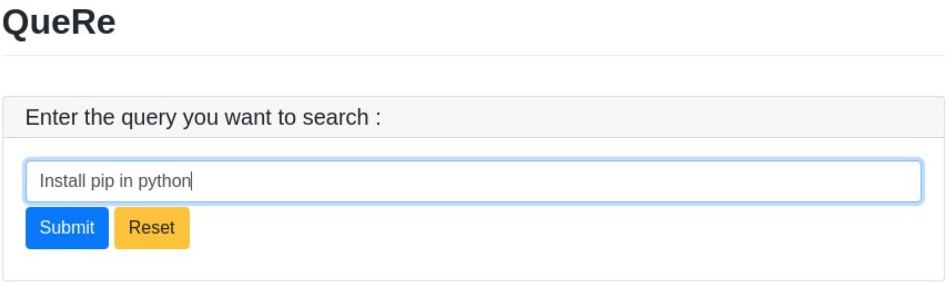
After the sentence embeddings are obtained, the cosine similarity score is computed for each archive sentence embedding with query sentence embedding using the cdist function. The question id of top 10 highest scored questions are retrieved.

4.6 Key Word based Matching

For key word based matching, we have used Elastic Search. Elastic Search create index of injected question and its question id in json format. Then Elastic Search uses the match query which has Lucene's default scoring function i.e., BM25 for keyword based similarity matching. BM25 similarity scoring is based on TF-IDF. TF-IDF calculates the importance of a word based on two properties i.e. proportional to term frequency and inversely proportional to document frequency. TF-IDF estimates the importance of a particular word in a single question as well as over the entire document collection.
Once the score of query against each question is calculated using the index, Then top 10 similar questions are retrieved along with their question ids.

4.7 Combined Ranking

Firstly the scores obtained by the key-word based matching is normalised to get in range of [0,1].
The scores of keyword based matching is multiplied by 0.3 weight and the scores of semantic based matching is multiplied by 0.7 weight. The question id are sorted in descending order on the basis of their updated scores. Finally, top 10 relevant question with their answer is returned using the extracted top 10 question id based on scores.

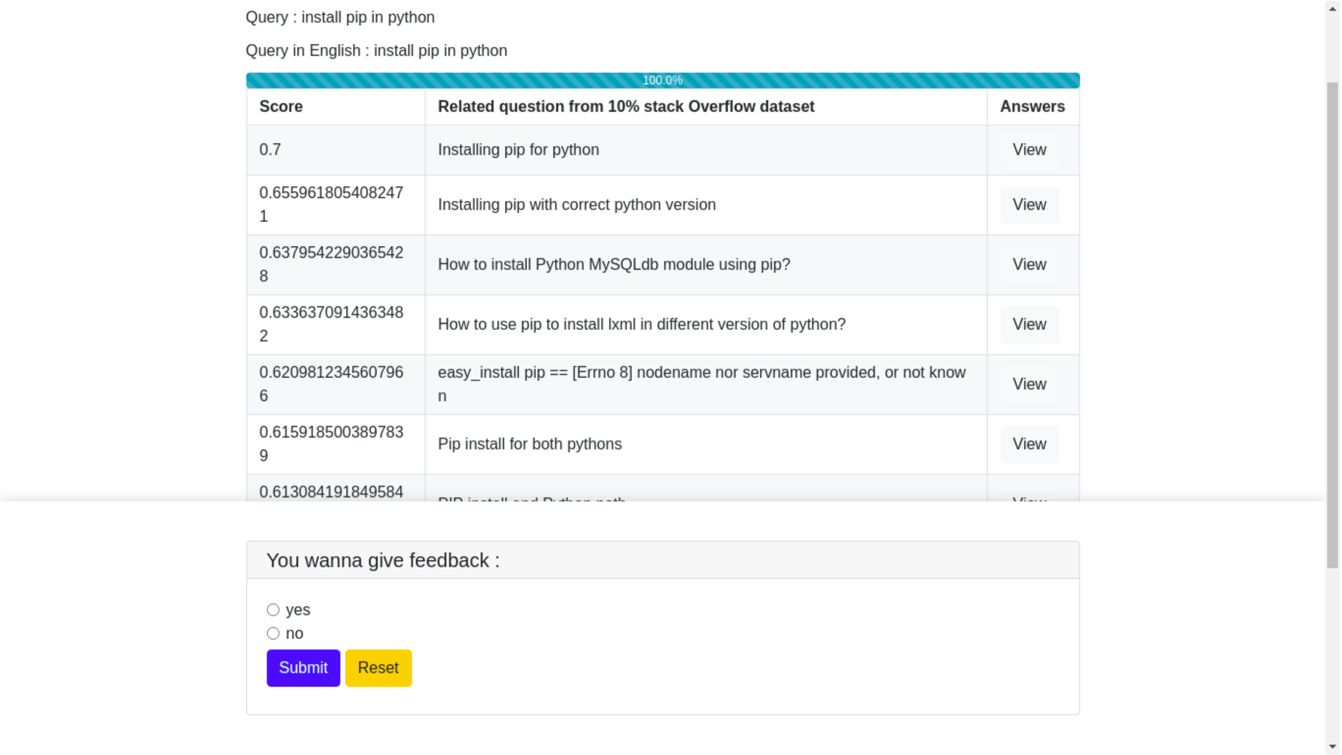


QueRe

Enter the query you want to search :

Install pip in python

Figure 2: Query Entered in English language



Query : install pip in python

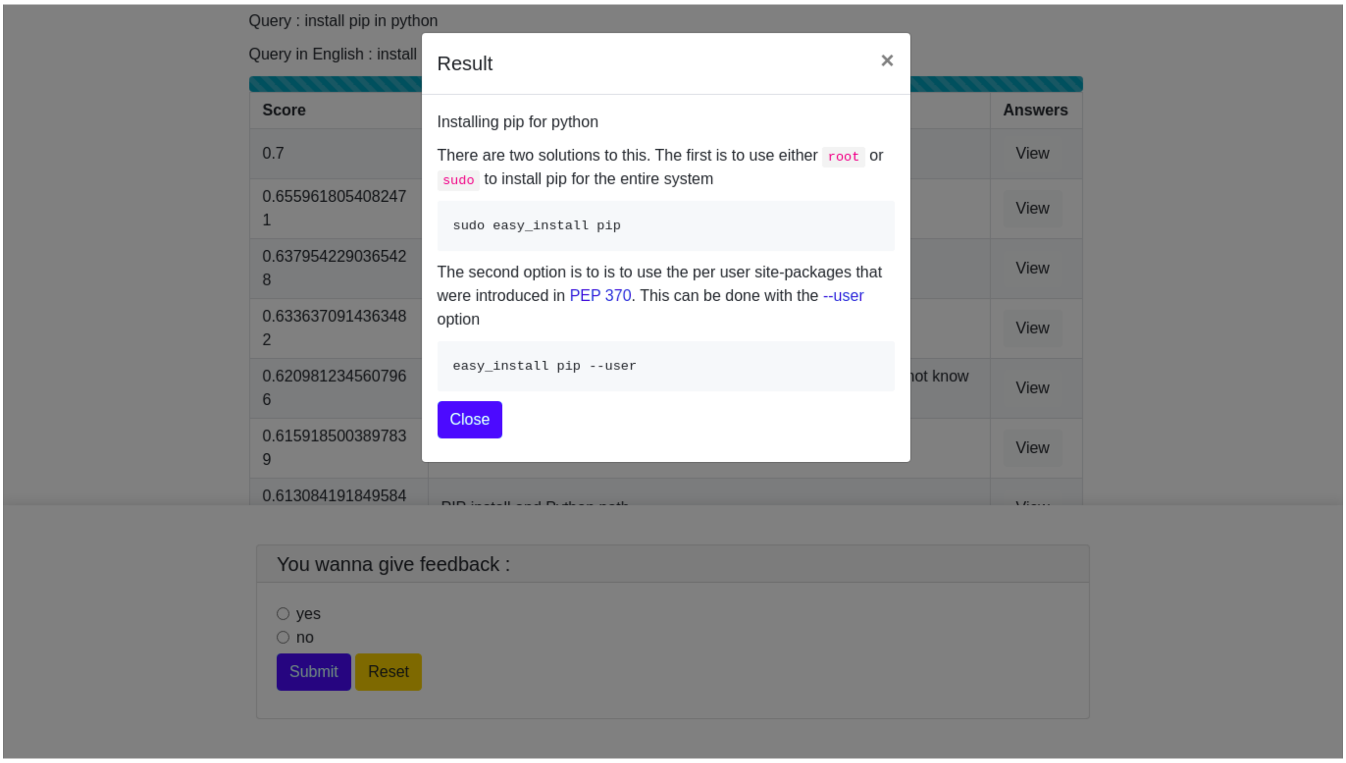
Query in English : install pip in python

Score	Related question from 10% stack Overflow dataset	Answers
0.7	Installing pip for python	View
0.6559618054082471	Installing pip with correct python version	View
0.6379542290365428	How to install Python MySQLdb module using pip?	View
0.6336370914363482	How to use pip to install lxml in different version of python?	View
0.6209812345607966	easy_install pip == [Errno 8] nodename nor servname provided, or not known	View
0.6159185003897839	Pip install for both pythons	View
0.613084191849584	PIP install and Python path	View

You wanna give feedback :

○ yes
○ no

Figure 3: Query Results



Query : install pip in python

Query in English : install

Score		Answers
0.7		View
0.6559618054082471		View
0.6379542290365428		View
0.6336370914363482		View
0.6209812345607966	not know	View
0.6159185003897839		View
0.613084191849584	PIP install and Python path	View

Result ×

Installing pip for python

There are two solutions to this. The first is to use either `root` or `sudo` to install pip for the entire system

```
sudo easy_install pip
```

The second option is to is to use the per user site-packages that were introduced in PEP 370. This can be done with the --user option

```
easy_install pip --user
```

You wanna give feedback :

○ yes
○ no

Figure 4: Question-Answer for the selected relevant question

5. EVALUATION

We used Precision@n (P@5 P@10) as the evaluation metric for determining the performance of our baseline model. Our model produced 0.5555 for P@5 and 0.49 for P@10 on the test queries. Figure 2 to 4 depicts the top-10 ranked questions along with their similarity scores for a given test query.

6. CONCLUSION

In our work, we showed how we build our model using the power of both , the cosine similarity using universal sentence encoders and key based Tf-idf models to retrieve the most similar and semantically related questions from our dataset. We also opened it for multiple languages (approximately it could handle 36 languages input) and published on web so that it could be accessible world wide on internet. In future we would like to work on the load balancing part of the web-app hypothesizing large traffic and also focus more on the categorized questions from forums and FAQ sites.

7. REFERENCES

[1] D. Cer, Y. Yang, S. yi Kong, N. Hua, N. L. U. Limtiaco, R. S. John, N. Constant, M. Guajardo-Céspedes, S. Yuan, C. Tar, Y. hsuan Sung, B. Strope, and R. Kurzweil. Universal sentence encoder. In *In submission to: EMNLP demonstration*, Brussels, Belgium, 2018. In submission.

[2] L. Chen, J. M. Jose, H. Yu, F. Yuan, and D. Zhang. A semantic graph based topic model for question retrieval in community question answering. WSDM '16, page 287–296, New York, NY, USA, 2016. Association for Computing Machinery.

[3] V. Efstathiou, C. Chatzilenas, and D. Spinellis. Word embeddings for the software engineering domain. In *2018 IEEE/ACM 15th International Conference on Mining Software Repositories (MSR)*, pages 38–41, 2018.

[4] X. Liu, C. Wang, Y. Leng, and C. Zhai. Linkso: A dataset for learning to retrieve similar question answer pairs on software development forums. In *Proceedings of the 4th ACM SIGSOFT International Workshop on NLP for Software Engineering*, NL4SE 2018, page 2–5, New York, NY, USA, 2018. Association for Computing Machinery.

[5] P. Nakov, D. Hoogeveen, L. Màrquez, A. Moschitti, H. Mubarak, T. Baldwin, and K. Verspoor. SemEval-2017 task 3: Community question answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 27–48, Vancouver, Canada, Aug. 2017. Association for Computational Linguistics.

[6] N. Othman, R. Faiz, and K. Smaïli. Using word embeddings to retrieve semantically similar questions in community question answering. *International Science and General Applications (ISGA journal 2018)*, 05 2018.

[7] N. Othman, R. Faiz, and K. Smaïli. Enhancing question retrieval in community question answering using word embeddings. *Procedia Computer Science*, 159:485–494, 2019. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 23rd International Conference KES2019.

[8] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *CoRR*, abs/1908.10084, 2019.

[9] W. Zhang, Z. Ming, Y. Zhang, T. Liu, and T. Chua. Capturing the semantics of key phrases using multiple languages for question retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 28(4):888–900, 2016.