Expert Systems With Applications 187 (2022) 115896

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A deep learning based end-to-end system (F-Gen) for automated email FAQ generation

Shiney Jeyaraj *, Raghuveera T.

Department of Computer Science and Engineering, College of Engineering Guindy, Anna University, Guindy, Chennai 600025, India

ARTICLE INFO

Keywords:
Expert system
Deep learning applications
Email text mining
Information retrieval
FAQ generation

ABSTRACT

With overwhelming volumes of official emails being exchanged in enterprises every day, emails have become vital information storehouses. Automatic generation of FAQs from email systems helps in identifying important information and could serve potential applications such as chatbots and intelligent email answering. While there exist studies in the literature focusing on automatic FAQ generation and automated email answering, there are few studies that apply recently developed deep learning techniques to fetch FAQs from emails. This paper proposes a novel framework named F-Gen, which is an expert system that generates potential FAQs from emails utilizing state-of-the-art methodologies. The key characteristics of this study are as follows: 1. Designing F-Gen with various subsystems that interoperate together for the FAQ generation 2. Identifying the parameters that determine a valid FAQ. The three subsystems of F-Gen are: (a) query classifier subsystem (QC subsystem) for email texts, (b) FAQ group generator subsystem (FGG subsystem) for generating FAQ groups from email queries. And (c) FAQ generator subsystem (FG subsystem) for conversion of email query clusters into FAQs. Experiments on the email dataset that practically reflect the above-mentioned problem resulted in FAQs with a ROUGE-1 F-Score of 74.10% when compared with the ground truth.

1. Introduction

In 2019, more than 293 billion emails were exchanged every day for business and official purposes, and this number is expected to exceed 347 billion by 2023 (Report, 2019). Official communication via email occurs to a large extent in organizations that interact extensively with clients. These organizations have a constant inflow of emails seeking information about their products or services. In addition, various customers/clients repeatedly send similar queries in heterogeneous contexts. The help desk or customer service teams patiently offer answers to these queries through email. However, at times this job can get frustrating for them as they have to respond to same queries multiple times and send redundant replies, whereas the organization also incurs wastage of resources such as time, cost, effort, and manpower. The generation of frequently asked questions (FAQs) from email repositories would mitigate the wastage of their resources to a larger extent. The generation of FAQs is traditionally done by domain experts, who identify potential FAQs from existing emails. They use their cognitive and language skills to rewrite the query so that it is coherent, complete, meaningful, relevant, grammatically correct, and not obsolete. Hence, they have an irreplaceable role in generating client-friendly FAQs. Moreover, their attrition costs the organization. This problem is depicted in Fig. 1 and has been previously discussed (Jeyaraj & Tripuraribhatla, 2018). However, to date, no study has transformed email content into valid and well-formed FAQs. In this study, we propose an F-Gen system that automates the work of a domain expert. The F-Gen system identifies potential FAQ groups (group of semantically similar queries) from the email content and generates the best possible FAQs (summary of FAQ group) from them. The FAQs generated by the system have practical applications in software services such as Microsoft's QnA Maker (Singh, Ramasubramanian, & Shivam, 2019), which uses the FAQ information for training chatbots. The FAQs can also be used in building enterprise IT management support systems (Bozdogan, Zincir-Heywood, & Zincir, 2015) as well as in answering inquiry emails (Itakura, Kenmotsu, Oka, & Akiyoshi, 2010). When we build the F-Gen system, the following research questions arise:

1. What are the key factors that determine whether or not an email text is a FAQ?
2. How do we develop an expert system that transforms email content into FAQs? What are the subsystems to be designed for that end?
3. How can we validate the automatically generated FAQs against the ground truth FAQ?

* Corresponding author.
E-mail addresses: shineyjeyaraj@gmail.com (S. Jeyaraj), raghuveera@annauniv.edu (Raghuveera T.).

https://doi.org/10.1016/j.eswa.2021.115896
Received 7 May 2020; Received in revised form 4 September 2021; Accepted 8 September 2021
Available online 22 September 2021

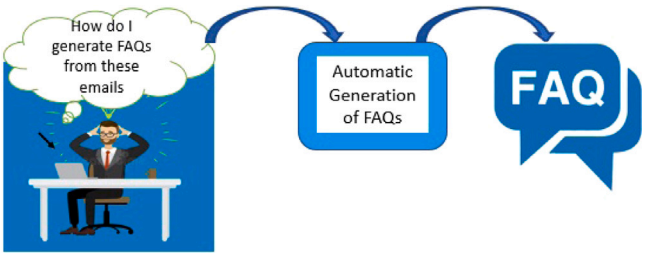

Fig. 1. Motivation behind F-Gen.

In this study, we try to find answers to these questions. We adopt deep learning techniques such as BERT-Pre-training of deep bidirectional transformers for language understanding (Devlin, Chang, Lee, & Toutanova, 2018) embeddings, Siamese LSTMs (Mueller & Thyagarajan, 2016), and the BERT summarizer (Liu, 2019). The contributions of this study are as follows:

1. Modeling FAQ and its characteristics
2. Building F-Gen system with interoperating subsystems

The remainder of this paper is organized as follows. Section 2 discusses the existing works related to information retrieval from emails and FAQ generation. Section 3 describes the conceptual model of an FAQ, and Section 4 explains our proposed F-Gen system. Section 5 details the experimental results, and Section 6 discusses the results. In Section 7, we present our conclusions and directions for future work.

2. Related work

In this section, we discuss and summarize the previous research related to our work. In Section 2.1, we list the various approaches for information extraction from emails, whereas in Section 2.2, we describe the studies related to computing semantic similarity across sentences. Section 2.3 lists studies that define the characteristics of a query.

2.1. Information retrieval from emails and helpdesk cases

One approach to information retrieval from emails is to search for past emails with the contents of a new incoming email. Naeem, Linggawa, Mughal, Lutteroth, and Weber (2018) designed a system that suggests replies to new emails by retrieving similar cases that exist in the past emails. They used text and semantic processing techniques to retrieve a better matching case. Case-based retrieval was a popular approach, and we studied the following two studies that performed case-based retrieval on helpdesk data. Wang, Li, Zhu, and Gong (2010) clustered cases similar to the new cases received in helpdesks. They also used many sentence-level semantics and retrieved similar past cases. Kang, Krishnaswamy, and Zaslavsky (2013) used association rule mining for the retrieval of similar cases. In case-based retrieval, the focus is on extracting parts of the email that are semantically close to the question in the upcoming email. Few other researchers have worked on extracting information from emails. One of the earliest classifications of email sentences based on speech acts was by Cohen, Carvalho, and Mitchell (2004), who classified sentences into any one of the classes-request, propose, amend, commit, and deliver. They defined their own ontology for each class and classified them based on that. In another study conducted by Sappelli, Pasi, Verberne, de Boer, and Kraaij (2016), the tasks within the email were extracted by observing the intent behind sending the email. Yelati and Sangal (2011) classified email sentences into any one of the following discourse segments: greet, goal, background, concern, query, and address. They handpicked lexical and semantic features to train an Support Vector Machine (SVM) classifier that could classify the email sentences into one of them. Yang, Awadallah, Khabsa, Wang, and Wang (2018) attempted to extract question and answer pairs from email threads. They developed a taxonomy of questions and classified the questions into different types. Kwong and Yorke-Smith (2009) were able to detect declarative and imperative questions from emails along with their answers. They used approximately 50 handcrafted patterns for the detection. Sneiders, Sjöbergh, and Alfalahi (2018) categorized email sentences into two categories: context description and response trigger. The context description describes the context in which the query was asked, whereas the response trigger is the sentence that seeks information. They used text-pattern matching for information retrieval. Alibadi and Vidal (2018) classified email sentences into two types: asking the receiver to do something (request, direct, or commit) and the other for the receiver to simply receive. The handpicked feature representation of the email sentences was replaced with word embeddings (GloVe, Word2Vec). The embeddings were fed into a convolutional neural network (CNN) and long short-term memory (LSTM) for classification. They achieved an F-measure of 0.91 and 0.85 for the two classes. The success of the email reply suggestion feature (Kannan et al., 2016) of Gmail indicates that a deep learning network along with the use of word embeddings is efficient in learning email contents. However, researches that use deep learning for the identification of queries in email are sparse. A few deep learning-based feature representation approaches for email text are highlighted in Sections 2.1.1 and 2.1.2.

Table 1
Comparison of text embedding models.

Embeddings	Context dependency	Granularity	Handles out of vocabulary words
Word2Vec (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013)	No	Word level	No
GloVe (Pennington, Socher, & Manning, 2014)	No	Word level	No
FastText (Joulin, Grave, Bojanowski, & Mikolov, 2016)	No	Character level	Yes
ELMo (Peters et al., 2018)	Yes	Character level	Yes
BERT (Devlin et al., 2018)	Yes	Subword level	Yes

2.1.1. Feature representation for email text

The generation of embeddings is either context-dependent or not. The embeddings we choose should preserve the contextual heterogeneity of the email text, give importance to domain-specific information, and handle out-of-vocabulary (OOV) words efficiently. Some of the prominent embeddings and their suitability for our email text are listed in Table 1. Among the context-dependent models, ELMo uses bidirectional LSTM for preserving contextual information, whereas BERT has a transformer architecture adopting an advanced masked language model, making it a powerful tool for representing sentence context. Word-level models cannot handle OOV words, whereas character-level models (Lee, Cho, & Hofmann, 2017) face drawbacks such as a longer training time and difficulty in handling long-range dependencies in input. Subword-level models strike a balance between the two. We chose a model that is context-dependent and works at the subword level. BERT works at subword level with its WordPiece tokenization and has been shown to be efficient in classifying text (Salminen et al., 2020). Furthermore, there are models that generate embeddings at the graph level (Narayanan et al., 2017) with graph to sequence learning (Beck, Haffari, & Cohn, 2018; Cai & Lam, 2020; Guo, Zhang, Teng, & Lu, 2019; Xu et al., 2018) approach gaining importance in recent times. However, choosing the right dimensionality is a challenge in graph representation learning (Goyal & Ferrara, 2018).

2.1.2. Feature representation when email is multilingual

Multilingual emails can be represented with pre-trained models such as mBERT, where m stands for multilingual (Devlin et al., 2018).

mBERT was trained using Wikipedia text in 104 languages. XLM (Lample & Conneau, 2019) and XLM-Roberta (Conneau et al., 2019) works with byte pair encoding and are even powerful in multilingual representations. Byte level representations (Gillick, Brunk, Vinyals, & Subramanya, 2015) of the input provides a compact language model that can handle multiple languages at a time. The model avoids the need for dedicated tokenizers for each of the different languages in the text. Languages such as Chinese have rich morphological variance and byte-level representation (Kenter, Jones, & Hewlett, 2018); even characters that were not seen during training are represented by a combination of bytes. A comparison between byte-level representation and character-level representation for the neural machine translation task shows that the training time decreases with byte-level representation (Ruiz Costa-Jussà, Escolano Peinado, & Rodríguez Fonollosa, 2017). Because the performance of byte-level models is directly proportional to the morphological complexity (Kenter et al., 2018) of languages, these models should be utilized when the input is multilingual (mainly with morphological variance).

2.2. Finding semantically similar questions

We created groups of semantically similar questions so that the FAQs could be generated. As we could not find many studies describing semantic similarity across questions/queries in emails, we gathered papers that find semantically similar queries in community question answering.

Zhang, Lo, Xia, and Sun (2015) considered features such as title of the question, its description, tag, and the topic to which it belongs for computing the similarity across question pairs. They experimented with the stack overflow data and observed that the recall rate was 10.2% better than that of previous studies. In another research, Zhang, Sheng, Lau, Abebe and Ruan (2018) proposed a method for detecting duplicates in programming community question answering sites. They used three types of features, namely, vector space distance, relevance of the question pairs, and presence of associations in the form of phrases between question pairs, and used approximately 404301 question pairs. They achieved a reasonable accuracy, which was nearly 25% better than the previous ones. As a result, the questions were classified as duplicate or relevant, with a maximum F-measure score of 0.694.

To determine whether a question pair is duplicate or relevant in a community question answering site, such as Stack Overflow, Zhang, Sheng, Tang and Ruan (2018) created correlation matrices for each question pair with the words in them. The correlation matrices were fed into a CNN for classification of the question pair. Mueller and Thyagarajan (2016) designed a Siamese LSTM network that learns from labeled data of duplicate sentences and predicts the sentence similarity for the new sentences. They experimented with the Sentences Involving Compositional Knowledge (SICK) dataset that contains approximately 9927 sentence pairs, resulting in an accuracy of 84.2%.

The study by Liang, Cui, Jiang, Shen, and Xie (2018) is closer to us as they built a FAQ system centered around queries about growing rice. They calculated the semantic similarity across questions in an FAQ system. They used Word2Vec embeddings with a CNN classifier, which resulted in an accuracy of 93.1%. Their dataset consisted of 3007 questions and 32,072 question pairs.

We see that the earlier approaches in the detection of duplicate or similar questions are mostly based on handcrafted features, and the latter are based on deep learning approaches. A study by Altheneyan and Menai (2020) states that deep learning techniques efficiently detect sentences that are rephrased. Deep learning techniques require labeled data. However, we do not have sufficient labeled data to determine whether the question pairs are similar. Detecting similar queries in the absence of a large amount of labeled data is challenging.

Table 2
Methods for query similarity computation.

Method	Advantages/limitations
Jaccard similarity	Does not take word order into account resulting in low precision
Cosine similarity	Does not take word order into account resulting in low precision
Word Mover's Distance (Kusner, Sun, Kolkin, & Weinberger, 2015)	Does not take word order into account resulting in low precision
LSTM (Hochreiter & Schmidhuber, 1997)	Capture the long term dependencies in text
Deep Siamese networks (Mueller & Thyagarajan, 2016) with LSTMs, CNNs and RNNs	Parallel Representation of queries makes it efficient in similarity computation

2.2.1. Techniques for semantic textual similarity

Semantic text similarity (STS) computation tasks previously used word level similarity, sentence composition, and other handcrafted features for training classifiers. Recently, word embeddings have been used for feature extraction. In Section 2.1.1, the advantages of context-dependent embeddings are discussed, and BERT embeddings are found to be best suited. However, context-dependent word embeddings generate similar embeddings for queries falling under the same context but differ semantically. Similarity computation using such embedding results in a large number of false positives because queries involving different entities with the same context are misclassified as similar. Hence, for similar query classification, state-of-the-art context-independent embedding, namely, Word2Vec (Mikolov et al., 2013) embedding, is preferred.

A comparison of the methods for the similarity computation is provided in Table 2. Siamese networks have identical substructures; specifically, the weights of the network are maintained the same on both networks and enable us to make parallel representations of the queries. Because LSTMs offer rich semantic representation of query pairs, we choose a Siamese network with LSTMs.

2.3. Characteristics of a good question

The FAQ generated by the F-Gen system should be of good quality, so that it is easily understood by anyone who reads the FAQs. Hence, we studied the literature to find studies that describe the characteristics of a good question. Kolomiyets and Moens (2011) define a question as a natural language sentence that begins with an interrogative word written by someone with an information need. They say that some sentences that are in imperative form might also request information. For automatic question generation from text, Heilman and Smith (2010) generated all possible questions and ranked them to find good-quality questions. They listed the deficiencies that a generated question might have, such as being grammatically incorrect, not making sense, being vague, having an obvious answer or missing answer, containing wrong wh-words, and having an improper format. According to Ravi, Pang, Rastogi, and Kumar (2014), a good question in the context of community question-answering means possessing clarity and the question being repeatedly asked over a long period of time. Rath, Shah, and Floegel (2017) have identified the traits of a bad question: being too broad, meaningless, general statement, written in a foreign language, having confusing words, socially awkward content, too many sentences or typos or garbled words, and missing information. We observe that clarity is an important characteristic of a question, and to achieve it, we ensure that the question generated contains the maximum possible relevant information. To generate such questions, we apply text summarization to a group of queries.

2.3.1. Summarization

To obtain the core idea of any text, automatic text summarization techniques were employed. Summarization can be either abstractive or extractive. The abstractive approach rewrites the actual content, whereas extractive summarization ranks sentences to extract the most important ones. For the generation of FAQs from FAQ groups, we prefer an extractive summarization approach. TextRank (Mihalcea & Tarau, 2004) was one of the standard graph-based extractive summarization techniques. Recently, the use of deep learning-based summarization, such as word embeddings (Alami, Meknassi, & En-nahnahi, 2019), auto encoders (Joshi, Fidalgo, Alegre, & Fernández-Robles, 2019), and BERT summarization (Liu, 2019), have been widely used. We have used the BERT-based summarization in this study.

3. Conceptual modeling

We model the problem of FAQ generation from emails by studying the characteristics of an FAQ. We depict the measures that determine the validity of the FAQ in the cause and effect diagram (Ilie & Ciocoiu, 2010) shown in Fig. 2. The analysis helps us define an FAQ, as in Definition 1.

Definition 1. A valid FAQ can be defined as a sentence that is in the form of a query, being frequently asked, complete, syntactically and semantically correct, and time appropriate.

Let us understand the underlying causes of an email sentence as a valid FAQ.

Cause 1. *To be a valid FAQ, a sentence has to be a query*:

There are three types of email sentences: one that (1) conveys personal information *I*, (2) describes contextual information *C*, and (3) asks for information *Q*. Sentences that talk about the name, designation, qualification, ethnicity, mobile number, identification number, place of working/ studying, date of joining/resigning, address, and other personal information are classified as Type *I* sentence. The second is the Type *C* sentence, which includes sentences that bring out the actual context or the need to write the email. The third type *Q* sentence is a query or request. *I, C,* and *Q* can occur in different sentences or can occur together in a sentence. In the case of emails with similar queries, *I* and *C* might be different, but the *Q* parts will be semantically similar. Hence, knowing the *Q* sentence beforehand will enable us to segregate it and find similarity across the *Q* sentences alone.

Previous researchers (Sneiders et al., 2018) have classified sentences into two types: context description and response trigger. They used pattern matching with predefined templates for the identification of Type *C* and Type *Q* sentences. In our study, we introduce Type *I* in addition to Types *C* and *Q* for classification because the FAQ generated with our method should not contain personal details. Hence, out of the three, only sentences of type *Q* are considered for FAQ generation. Each email *E* is represented as a function of *I, C,* and *Q* occurring in varying proportions, as shown in Eq. (1).

$$E = \{I, C, Q\} \tag{1}$$

Cause 2. *To be a valid FAQ, a query has to be frequent (F)*:

Queries extracted from multiple emails might be similar/duplicates or not. The similarity can be due to the presence of similar keywords or textual semantic similarity across the queries. So, out of all the queries only duplicate queries above the frequency threshold f can qualify to be a *FAQ*. Let *SimGrp* be a group of n similar queries as denoted in Eq. (2). The *SimGrp* being frequent or not is represented in Eq. (3).

$$SimGrp = \{Q_1, Q_2 \ldots Q_n\} \tag{2}$$

$$F(SimGrp) = \begin{cases} 1, & \text{if } n > \text{f} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Cause 3. *To be valid FAQ, a frequent query has to be Complete(CP)*:

When there is a group of semantically overlapping queries, few queries might possess a higher amount of information compared with the others. For example, consider the queries, *"The files were uploaded but I cannot find them"* and *"I cannot find the file"*. Both convey similar meanings, but the first query is highly informative compared with the second one. Furthermore, there can be sentences such as *"I want to know how to download multiple files"* and *"I want to download file multiple times"*. Both are not exactly similar; hence, we would like to integrate both to obtain an optimal summary that is represented by a single complete query.

The completeness of a query is a relative term and we measure it by comparing its domain information index *DII* with others, as shown in Eq. (4).

$$CP(Q_i) > CP(Q_j) \text{ if } DII(Q_i) > DII(Q_j) \tag{4}$$

Cause 4. *To be a valid FAQ, a complete frequent query must be correct(CI)*:

The readers of an FAQ might get annoyed and confused if there are errors or grammatical errors in the query. Incorrect spelling, improper grammar usage, and lack of coherence could be the potential causes of incorrect queries. We measured the level of correctness of a query using the correctness index *(CI)*. We show the relationship between *(CI)* and *Err* (the amount of spelling and grammatical errors) in Eq. (5).

$$CI(Q_i) \propto \frac{1}{Err(Q_i)} \tag{5}$$

Cause 5. *To be a valid FAQ, a correct, complete, and frequent query must be time appropriate (TA)*:

The nature of FAQ varies with time. A query that is frequent at one point in time does not always need to be frequent. These queries are eligible for FAQs. For example, queries regarding admission to an institution are frequent every year during the admission season. However, certain queries that have become obsolete are not eligible for FAQs. For a valid FAQ, the value of TA must be 1, as in Eq. (6).

$$TA(Q_i) = 1 \tag{6}$$

We modeled the valid FAQ in Eq. (7) for queries that satisfy Eqs. (3)–(6).

$$FAQ = \{\forall Q \mid (F, CP, CI, TA) \in \{\mathbb{R}\}_{>0}\} \tag{7}$$

4. The proposed F-Gen system

The F-Gen system performs varied functions in each subsystem to achieve a single goal (FAQ Generation). It starts from email-specific pre-processing and ends with the generation of valid FAQs, as modeled in Section 3. Email specific pre-processing includes extraction of emails from eml files followed by removal of all email-specific stop words such as salutations, words like regards, yours truly, etc., and any personal information. The subsystems operate in a sequential pattern receiving input from the previous one and feeding the output to the next one. The overall system architecture is illustrated in Fig. 3.

F-Gen currently handles only English language emails. This is because the embeddings for feature representation (BERT, Word2Vec) were pretrained on the English corpus. To handle multilingual emails, one of the techniques discussed in Section 2.1.2 has to be employed.

4.1. Query Classifier subsystem (QC subsystem)

In this subsystem, a novel classifier is designed to classify the email sentences into *I, C,* and *Q* types, as mentioned in Section 3. It receives email sentences as the input and outputs the queries to the next subsystem. The email sentences are represented with their feature vectors, which are later used for training the classifier models. To derive feature vectors for the query classifier, the BERT model was chosen because of its advantages mentioned in Table 1. Because our dataset is in English, we chose BERT over multilingual and byte-level models.

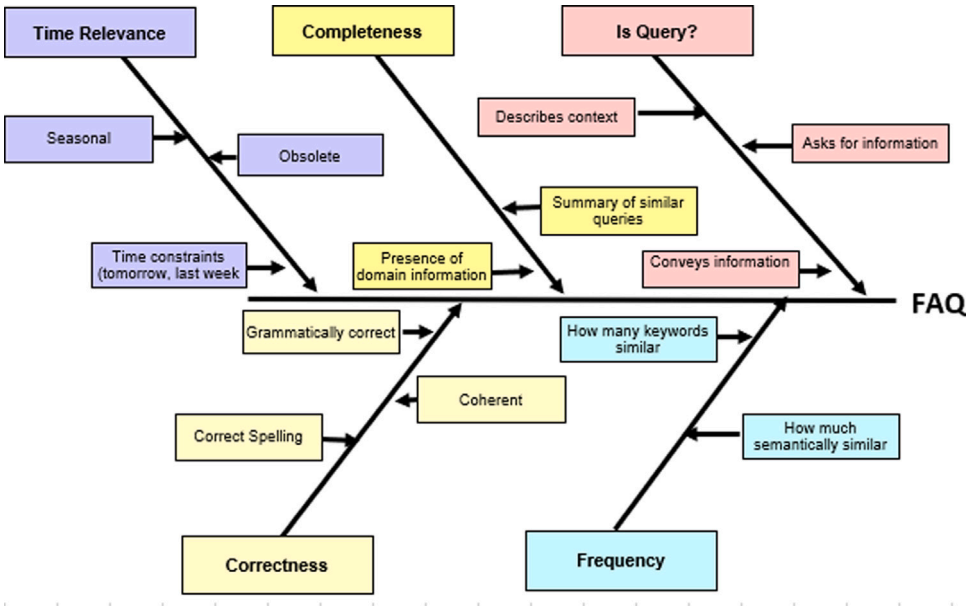

Fig. 2. Factors that contribute towards a Valid FAQ using Fishbone diagram.

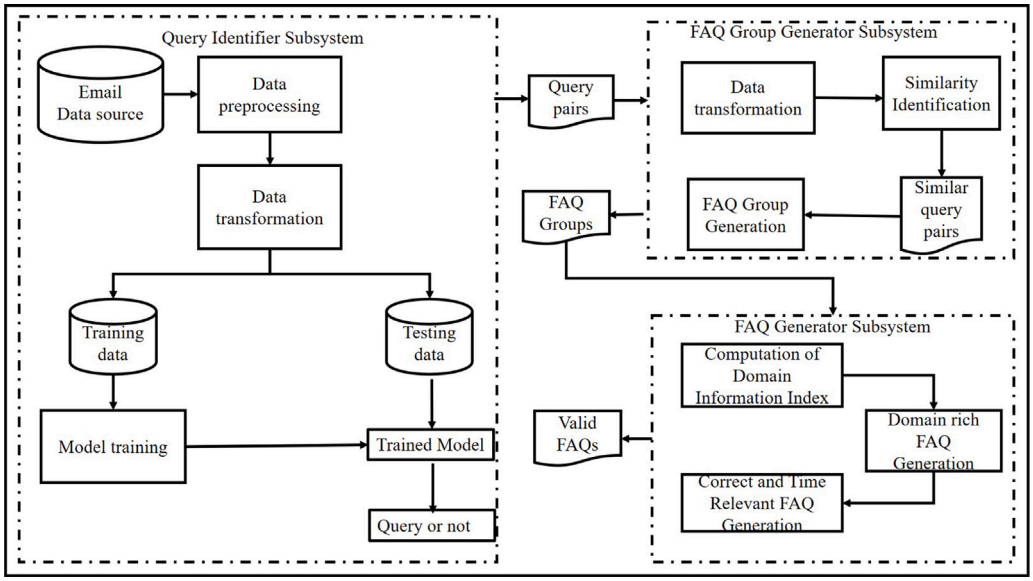

Fig. 3. Architecture of the proposed F-Gen System.

Feature extraction using BERT. The pre-trained BERT model (base version with 12 hidden layers, 768 hidden units, and $110M$ parameters) was loaded into a transformer encoder, and the pre-processed sentences were fed into it. In Fig. 4, the embedding generation of a sample type I input sentence is described. The sentence begins with *[CLS]* token, and separator *[SEP]* indicates the end of the sentence segments. For each token T, token embeddings (T_{TE}), segment embeddings (T_{SE}), and position embeddings (T_{PE}) are generated and combined (T_{Emb}), as shown in Eq. (8). Sentence S with n tokens generates embeddings of dimensions $n * 768$. The T_{Emb} embeddings corresponding to *[CLS]* for each S were extracted as feature vectors.

$$T_{Emb} = T_{TE} + T_{SE} + T_{PE} \tag{8}$$

BERT. The position embeddings distinguish the tokens based on their position of occurrence in sentences, while segment embeddings differentiate tokens among various sentence segments. As a result, the context in which tokens occur contributes to combined embedding. Generally, queries and non-queries have the same tokens occurring in different positions of a segment or in various segments; thus, the combined token embeddings of BERT intuitively best represent the input for query classification compared to context-independent embeddings.

Query classifier. The feature vectors along with their sentence labels (I, C, and Q) are split into training and testing sets. Four classifiers, namely, NaiveBayes (Lewis, 1998), LDA (Ramage, Hall, Nallapati, & Manning, 2009), SVM (Joachims, 1998), and k-nearest neighbors (Cover & Hart, 1967) were individually trained. The SVM classifier with linear kernel proved to yield better results and was chosen as suitable for the query classification task.

4.2. FAQ Group Generator subsystem (FGG subsystem)

The potential queries have been identified in Section 4.1, and the next step is to group the semantically similar queries so that the frequently asked queries can be retrieved. Let us assume a list of

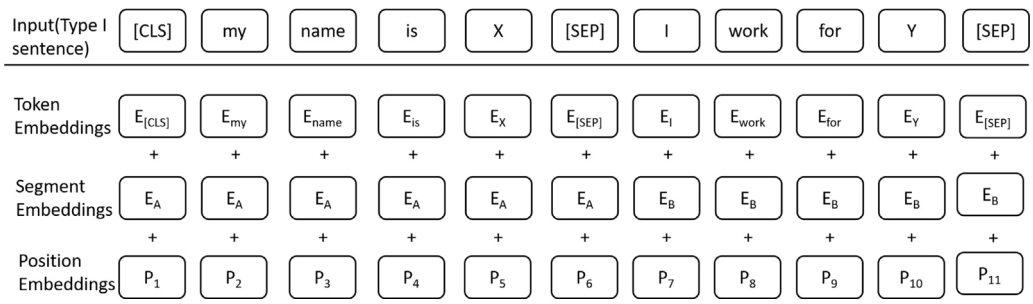

Fig. 4. BERT embedding generation for a sample sentence.

queries $QList = \{Q_1, Q_2 \ldots Q_n\}$. All possible combinations of queries $C_k = \{Q_i, Q_j\}$ are generated such that i ranges from *1* to n, j ranges from *1* to n, and $i < j$. Combinations C_k in $Qlist$ form the input to the subsystem that generates FAQ groups. Word2Vec embeddings of the query pairs in the combination list C_k are fed as input into two parallel networks, as shown in Fig. 5. If the output from the LSTM networks for query Q_i is O_i and that for query Q_j is O_j, then the similarity is predicted using the Manhattan similarity in Eq. (9). The Siamese network parameters are optimized with the ADAM (Kingma & Ba, 2014) optimizer, which best suits us because the training dataset is large. A dropout of 0.2, was added to the LSTM layers to prevent over-fitting of the network.

$$Sim(Q_i, Q_j) = e^{-|O_i - O_j|} \tag{9}$$

4.2.1. Transfer learning for computing similarity

A large number of training sentence pairs are required to predict sentence similarity using the Manhattan LSTM. The publicly available Quora Question Pairs Dataset (Dataset, 2017), which has over 400,000 pairs of questions with labels, $is\_duplicate = 0$ *or* 1, is used as the training set.

4.2.2. Similarity thresholding

The percentage of similarity across query pairs is critical in determining the FAQ groups. A high similarity threshold s results in high precision and low recall. However, a low s results in low precision and high recall. For experimentation, s was varied between 0.75 and 1.0. Query combinations C_k with similarity above the threshold are stored in $SimQList$, while the others are discarded.

4.2.3. Context dependent queries

A few queries vary semantically, depending on the context. Consider two queries: *Q1: What is the procedure to apply for a PhD? Q2: What is the procedure to apply for PhD at XYZ University?* Here, $Q1$ may refer to the application procedure at *XYZ University* or a totally different one. They are predicted to be similar or dissimilar depending on the choice of s. The value of s is optimally fixed such that a good precision–recall trade-off is achieved. Determining the optimal threshold is dataset specific and, hence, a limitation of F-Gen. In the future, these queries may be handled with domain-adapted embeddings (Sarma, Liang, & Sethares, 2018), which is a combination of generic embeddings (trained on a large generic corpus) and domain-specific embeddings (trained on domain-specific datasets). Alternatively, the semantic matching model (Xu & Yuan, 2020) performs well because it gives attention to similar words only when there is semantic similarity across sentences. Another approach is to compute similarity by taking their word embeddings in addition to knowledge from external sources (Nguyen, Duong, & Cambria, 2019).

4.2.4. Generation of FAQ groups

The similarity of queries shows transitive dependency, as stated in Definition 2. Let us consider the $SimQList$ sample, as shown in Eq. (10), and group its members based on Definition 2. The result of the grouping operation on $SimQList$ is $SimQList_{new}$, as shown in Eq. (11). After all possible groupings, $SimQList$ is replaced with members of $SimQList_{new}$. The frequency threshold f is fixed ($f = 5$ for experimentation), and members of $SimQList$, which have at least f items in them, are chosen and added to the *FAQGrpList*.

Definition 2. If Q_i is similar to Q_j and Q_j is similar to Q_k, then by transitive dependency, Q_i, Q_j and Q_k are similar.

$$SimQList = \{(Q_1, Q_2), (Q_1, Q_3), (Q_1, Q_4), (Q_4, Q_5), (Q_6, Q_7)\} \tag{10}$$

$$SimQList_{new} = Group(SimQList) = \{(Q_1, Q_2, Q_3, Q_4, Q_5), (Q_6, Q_7)\} \tag{11}$$

4.3. FAQ Generator subsystem (FG subsystem)

The eligible FAQ groups in the *FAQGrpList* consist of queries that are frequently asked. This subsystem generates valid FAQs according to the model in Section 3 from *FAQGrpList*.

4.3.1. Generating domain rich FAQs using domain based summarization

We incorporate the BERT extractive summarization because it learns the context of sentences effectively through combined word embeddings, as discussed in Section 2.1.1. In BERT-based summarization, the sentence embeddings are generated by combined embeddings, as described in Fig. 4, and fed into summarization layers (inter-sentence transformers) to obtain the document-level significant features. Transformers internally encode and decode operations that give attention to the important features while forgetting others. Consider, a BERT sentence embedding, T. When the position embeddings are added to T, it is denoted by h_0. The operations in the summarization layers are shown in Eqs. (12) and (13), where h^l represents the output of a layer l, $LayerNorm$ denotes normalization function (Ba, Kiros, & Hinton, 2016) and $MultiheadAtt$ indicates multi-head attention (Vaswani et al., 2017).

$$\tilde{h}^l = LayerNorm(h^{l-1} + MultiheadAtt(h^{l-1})) \tag{12}$$

$$h^l = LayerNorm(\tilde{h}^l + FeedForwardNet(\tilde{h}^l)) \tag{13}$$

Domain rich summarization. We design a transformation step before summarization to obtain summaries with maximum possible domain information and call it domain rich summarization. Higher weights were assigned to queries with a higher domain information index (DII). To calculate the number of domain words, topic modeling of the corpus using the non-negative matrix factorization (NMF) (Lee & Seung, 2001) method is used. Let Df be the domain file where all

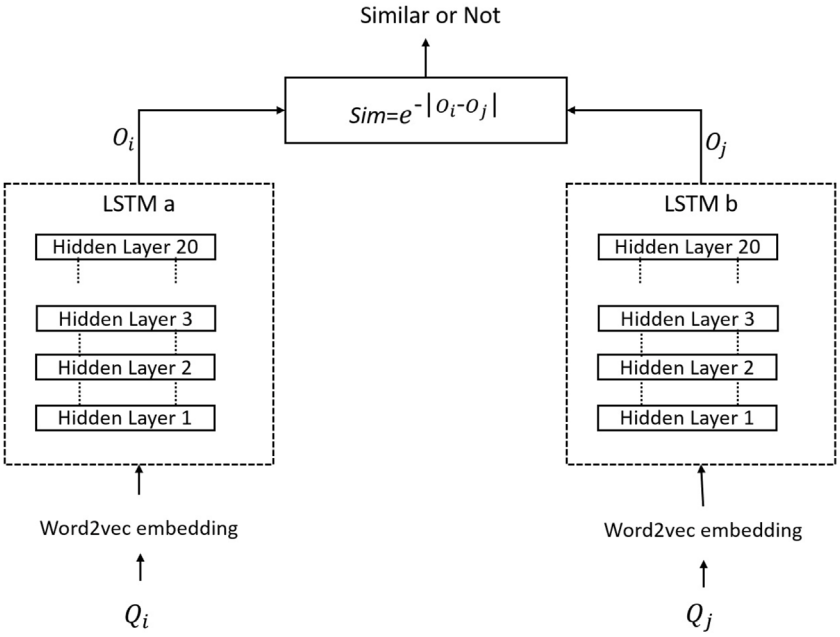

Fig. 5. Siamese LSTM for sentence similarity prediction.

Pseudocode 1 High Level Pseudocode for the extraction of FAQ groups from emails

```
Output:Probable FAQ Groups
Input: List of input emails, E = {E_1, E_2.....E_n}
List of queries, QList = {}
List of query combinations, C = {}
List of query combinations with similarity, SimQList = {}
List of FAQ Groups, FAQGrpList = {}
Similarity threshold, s
Frequency threshold, f
Query, Q
for E_i in E do
    Preprocess E_i
    Split E_i into sentences S_i = {S_i1, S_i2.....S_im}
    for S_ij in S_i do
        if isquery(S_ij) == 1 then
            append S_ij to QList
        end if
    end for
end for
for Q_i in QList do
    for Q_j in QList do
        if i < j then
            if compute similarity(Q_i, Q_j) > s then
                append C_k = (Q_i, Q_j) to SimQList
            end if
        end if
    end for
end for
while Grouping is possible do
    for SimQList_i,SimQList_j in SimQList do
        if same Q in SimQList_i&SimQList_j then
            combine SimQList_i&SimQList_j into SimQList_new
            append SimQList_new to SimQList
            delete SimQList_i&SimQList_j from SimQ_List
        end if
    end for
end while
for SimQList_k in SimQList do
    if count(members(SimQList_k)) > f then
        append SimQList_k to FAQGrpList
    end if
end for
```

Pseudocode 2 High Level Pseudocode for generation of valid FAQs

```
Output:Probable FAQs
Input: List of FAQ Groups, FAQGrpList={F_1, F_2...F_n}
List of FAQs, FAQs = {}
List of Complete FAQs, CompleteFAQs = {}
for F_i in FAQGrpList do
    for Q in F_i do
        Calculate DII
        Sort all Q in F_i based on descending DII
    end for
    Append Extractive summarization (F_i) to CompleteFAQ
end for
for Q in CompleteFAQs do
    CI=SpellAndGrammarCheck(Q)
    TA=TimeAppropriateCheck(Q)
    if CI == 1 & TA == 1 then
        Append Q to FAQs
    end if
end for
```

the domain words are stored. Consider a query Q with n number of words, excluding stop words. Let d be the number of words in Q that are also present in the Df. The DII of Q is calculated using Eq. (15).

The weighted query set is given by Eq. (14).

$$Q_{weighted} = \{(Q_1, DII_1), (Q_2, DII_2)....(Q_n, DII_n)\} \tag{14}$$

$Q_{weighted}$ is sorted in descending order of their DII values as Q_{sorted}, which forms the input to the BERT single document extractive summarizer. The reason for sorting is that the summarizer gives the utmost importance to the first query among other queries of the set. As a result, concise and domain rich FAQs were generated from each of the FAQ groups.

$$DII(Q) = \frac{d}{n} \tag{15}$$

4.3.2. Ensuring correctness and time appropriateness in FAQs

The generated queries should be grammatically correct and devoid of any errors. The spelling and grammar of FAQs is checked using existing tools(Python's pyLanguagetool). Let m be the number of spelling errors, and n be the number of grammatical errors. The correctness index (CI) of FAQ is defined in Eq. (16).

$$CI(Q) = \frac{1}{m+n+1} \tag{16}$$

When m and n were 0, the CI value was 1. As the value of m or n or both increases, CI decreases. If the CI is below a certain correctness

threshold (the minimum CI that is tolerable), then an FAQ that needs manual intervention is generated. To determine whether a query is time appropriate, we follow simple heuristics. A *time-appropriate query* is a contemporary one and is determined by three factors: *ageing, time-dependent,* and *non-seasonality*. *Ageing* is the process by which a query that is frequent at one point in time has become infrequent. *Non-seasonality* indicates that the query becomes frequent during a season and is infrequent in the next season. Measuring *ageing* and *non-seasonality* is beyond the scope of our study. We focus on the removal of time-dependent queries that are valid in an email but become invalid when put on an FAQ page. Hence, we define the list of time-dependent TR words and phrases (now, today, tomorrow, next day, next month, next year, previous week, this month, this Monday, etc.). If the FAQ contains such TR words and phrases, then the time-appropriate index TA is assigned as 0. Otherwise, the value of TA was 1. To summarize the work done, we identified possible queries from emails and checked whether they were eligible FAQs so that the FAQs generated by the system significantly matched the manually generated FAQs. The high-level pseudocode of our work is detailed in Pseudocode 1 and Pseudocode 2.

5. Experimental evaluation

5.1. Dataset

A real-time email inbox that reflects the problem in hand with 300 emails was chosen for experimentation. After extracting the body of the email and removing salutations, we obtained 1534 sentences: 383 were manually annotated as class I, 503 as class C, and 648 as class Q. In addition, a list of FAQs was manually crafted and represented the ground truth (G). To compare our work in Section 5.4.3, the English version of the FAQ dataset that was released for the purpose of training chatbots (Sumikawa, Fujiyoshi, Hatakeyama, & Nagai, 2019) was used. The dataset was originally created in Japanese and translated into English. The FAQ dataset consists of 79 FAQ groups, and each group is represented by a single ground truth (FAQ) that summarizes the FAQ group. The FAQ dataset is similar to our dataset in representing the FAQ groups, except that the queries were collected from an eLearning system of a Japanese university rather than an email mailbox. We elaborate upon the experimental results in Section 5.4.

5.2. Evaluation standards

All experiments (including those of previous authors repeated for comparison purposes) were conducted using Python libraries such as Keras, TensorFlow, and Sklearn. In the QC subsystem, ten-fold cross-validation is used for estimating classifier accuracy. The FGG and FG subsystems have custom test sets, and the evaluation is performed by comparison with the ground truth.

5.3. Performance metrics

5.3.1. QC Subsystem

The QC Subsystem evaluates classifier performance in terms of the F-Score in addition to accuracy. The F-score is calculated in Eq. (19) from the precision and recall defined in Eqs. (17) and (18). True Positive (TP) for class I is the number of sentences that are classified by the classifier as class I, whose ground truth is also class I. False Positive (FP) of class I is the number of sentences that are classified as class I, whose ground truth is not class I. False Negative (FN) for class I is the number of sentences that are classified as class I but whose ground truth is class I.

$$Precision = \frac{TP}{TP + FP} \tag{17}$$

$$Recall = \frac{TP}{TP + FN} \tag{18}$$

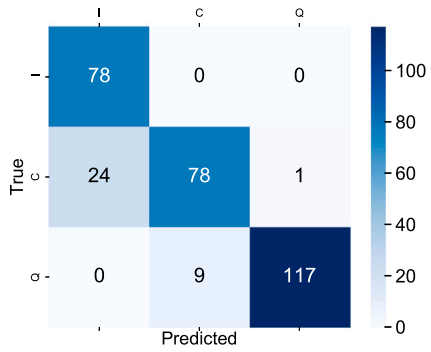

Fig. 6. Confusion matrix of SVM classifier obtained by averaging confusion matrices of all ten fold cross validation runs.

$$F - Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{19}$$

5.3.2. FGG subsytem

This grouping subsystem defines the grouping precision (GrP), grouping recall (GrR), and grouping F-score as in Eqs. (20)–(22), respectively

$$Grouping\ precision(GrP) = \frac{\text{No of FAQ groups correctly identified by system}}{\text{Total number of FAQ groups identified by system}} \tag{20}$$

$$Grouping\ recall(GrR) = \frac{\text{No of FAQ groups correctly identified by system}}{\text{Total number of FAQ groups in the ground truth}} \tag{21}$$

$$F - Score = \frac{2 * GrP * GrR}{GrP + GrR} \tag{22}$$

5.3.3. FG subsystem

The $ROUGE$ (Lin, 2004) metrics were used to compare the generated FAQs against the ground truth. Using ROUGE-1 and ROUGE-2, the average values of precision, recall, and F-Score were calculated.

5.4. Experimental results

5.4.1. QC subsystem

As discussed in Section 4.1, classifier models are trained using the BERT embeddings of email sentences. The confusion matrix of the SVM classifier is shown in Fig. 6, and all evaluations were performed with ten-fold cross validation. The classifier accuracy was 88.60% and the F-Score was 94.47%.

5.4.2. FGG subsystem

With 648 queries, $648C_2 = 2,09,628$ query combinations were generated. To find semantically similar queries, a Siamese LSTM network with 20 layers was designed. The maximum length of the input queries was set to 20, and the inputs were provided in batches of size 512. The model was trained for 50 epochs with 256 dimensional Word2Vec vectors of 4 000 000 question pairs with labels ($is\_duplicate = 1$ or 0). Using the trained model, the semantic similarity of $2,09,628$ query pairs was predicted. A few pairs of queries with their semantic similarities computed by the Siamese LSTM are listed in Table 3. When the semantic threshold was set to 0.827, 966 pairs of semantically similar queries were obtained and further grouped. 6 out of 8 (ground truth) FAQ groups were generated when the frequency threshold was set to 5. The other groups had fewer than five queries and were discarded. We

Table 3
Few query pairs along with their similarities calculated by FGG subsystem.

Query 1	Query 2	Similarity
Could you please reset my Progress Report?	I have forgotten my password for progress report.	0.8269431
Am I eligible to write the exam? Or I need to produce any other.	I have registered with this email id and password but it is not working now	0.8211477
When will the written test and interview be?	When will the written test and interview be.	1
Should I need to fill and submit the registration form?	Please reply if any issue in my application form	0.8504299
Should I need to fill and submit the registered form?	I had registered with this mail id and password but it is not working now.	0.80013114

Table 4
A sample FAQ group generated by F-Gen system (Email dataset).

S. No	Queries in FAQ group
1	I mail this to enquire about the admission procedure for JanuaryYYYY.
2	So please kindly tell me what to do and how to apply?
3	Please do inform me what is the procedure to apply for Z.
4	Am I eligible for applying to Z?
5	Can you please tell me from which option my supervisor has to approve my application?

Table 5
Domain rich words identified by NMF under few topic groups.

Topic group	Top keywords identified by NMF based topic modeling
1	Report progress scholar reg office check research extension thesis semester
2	Journal annexure paper synopsis publication list acceptance research submit accepted

obtained a grouping precision of 100%, grouping recall of 75%, and hence, the F-score was 85.71%. A sample FAQ group is presented in Table 4.

5.4.3. FG subsystem

The domain rich words in the dataset identified by NMF for the two sample topic groups are shown in Table 5. The calculated DII scores ranged from 0 to 6 across the queries, and the queries were reordered in each FAQ group with descending DII values. The reordered queries after BERT summarization were further checked for spelling and grammar mistakes using the Python built-in tools. In addition, the summary was checked for time appropriateness. Finally, a list of six FAQs was generated for the email dataset resulting in a ROUGE-1 F-Score of 74.10 and ROUGE-2 F-Score of 60.63. The experiment was also performed on the FAQ dataset and resulted in ROUGE-1 and ROUGE-2 F-Scores of 82.35, and 74.02, respectively. The FAQ sample generated by our approach is shown in Table 6.

6. Discussion

This section analyzes the classifier performance in the QC subsystem, FAQ grouping performance in the FGG subsystem, and compares the FAQ quality generated by the FG subsystem against BERT summarization.

6.1. Performance of QC subsystem

The performance of various classifiers was analyzed to determine the best fit. With ten-fold cross validation, a boxplot of the accuracy of the classifiers is shown in Fig. 7. It can be observed that although Naive Bayes performs the best, on an average scale, SVM with a linear kernel

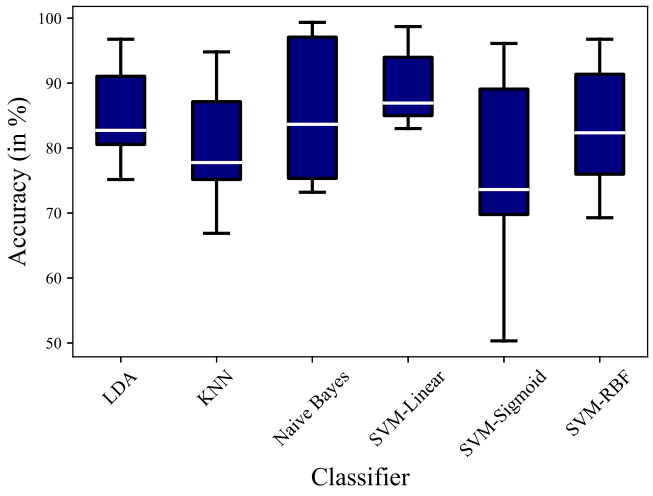

Fig. 7. Accuracies of different classifiers for query classification.

provides better results. The training time of the classifiers in Fig. 8 shows that although the training time is higher for SVM, it catches up with other classifiers in terms of time for testing. Considering the trade-off between time and accuracy, the classifier with the highest accuracy was selected. Experiments were conducted by varying the feature extraction methods for text such as *Word2Vec, GloVe, FastText, ELMo* and *BERT*, with their pre-trained models listed in Table 7. The classifier (SVM) accuracy with each of these embeddings is compared in Fig. 9 and it is noticed that BERT outperforms others in extracting features of the email dataset for query classification.

The receiver operating characteristic (ROC) curve is an optimal measure of the classifier performance. The highest area under the curve (AUC) was obtained for SVM with linear kernel(0.94), as shown in Fig. 10(d). In addition, we compare it with a previous email text classifier by Sneiders et al. (2018) in Table 8.

6.2. Performance of FGG subsystem

When the Word2Vec embeddings of query pairs are generated and their similarity is calculated using Word Mover's distance, the precision is only 29.04%, even when the minimum distance is fixed as a threshold. This indicates that there are queries containing similar words with multiple meanings, resulting in false positives. In contrast, Siamese LSTMs attain reasonable precision and recall, as shown in Fig. 11.

To compute the similarity across query pairs, it is necessary to set a suitable similarity threshold. The precision–recall graph in Fig. 11 shows a decrease in precision and an increase in recall at various points when the threshold is lowered from 1 to 0.80. The optimal threshold (0.827, in our case) is chosen such that there is a good trade-off between precision and recall. If the precision is poor, it affects the quality of the generated FAQs, whereas low recall would miss potential FAQs.

6.3. Performance of FG subsystem

The proposed domain rich summarization yields better F-Scores in terms of ROUGE-1 and ROUGE-2 compared with the existing BERT (Liu, 2019) summarization. Domain rich summarization as well as BERT summarization were experimented on our email dataset and FAQ dataset in Fig. 12.

6.4. Observations

We list five error measures along with their causes in Table 9. Misclassification occurs in the QC subsystem either because the query is

Table 6
A sample FAQ generated by F-Gen on FAQ dataset.

FAQ group	Ground truth	FAQ using BERT summarization	FAQ using proposed method
Can I set a password for documents? Does kibaco have encryption capabilities? I want to set the file to open with password authentication. Can I upload encrypted files? Can I upload files with password settings?	Can I upload encrypted files with password settings on kibaco?	Can I set a password for documents? Can I upload files with password settings?	I want to set the file to open with password authentication. Can I upload files with password settings?

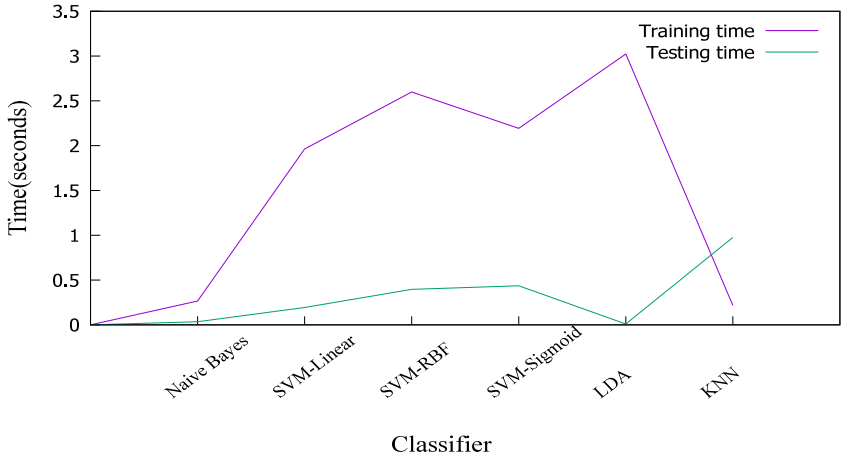

Fig. 8. Average time taken for training and testing different classifiers for query classification.

Table 7
Feature extraction methods and their implementation details for query classification.

Feature extraction method	Pre-trained model	Open source library/repository
Word2Vec	'GoogleNews-vectors-negative300.bin'	Gensim
GloVe	'glove.6B.300d.txt'	Gensim
FastText	'cc.en.300.bin'	fastText
ELMo	'tfhub.dev/google/elmo/2'	Tensorflow Hub
BERT	'bert-base-uncased'	'https://huggingface.co/models'

Table 8
Comparison of proposed method(F-Gen) with an existing approach for query classification.

	Proposed method (F-Gen)	Sneiders et al. (2018)
Classes	Identification(I), Context(C), Query(Q)	Context description, response trigger
Dataset	Custom	Custom
Features	Word embeddings	Handcrafted
Feature extraction	Unsupervised deep learning	Pattern matching
Classifier	SVM	SVM

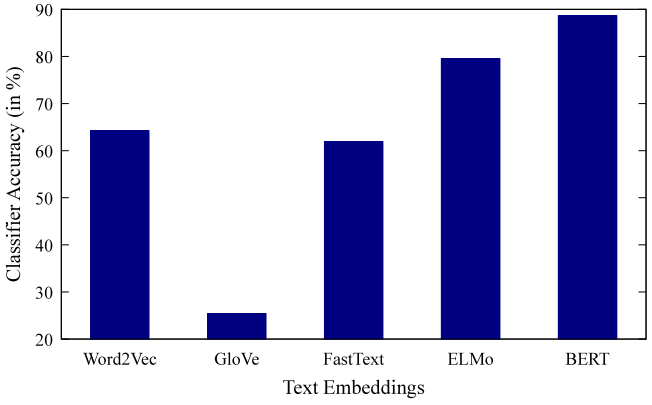

Fig. 9. Accuracies of SVM classifier with different text embeddings for query classification.

fuzzy or implicit. An example of fuzziness is as follows: *"I am X working for Y and would like to know the date of the test"*. The sentence belongs to both classes *I* and *Q*. An example of an implicit query is, *"I cannot take the test"*. The writer of this implicit query expressed his or her inability to take the test with the intention of knowing how to take the test. The absence of domain-specific wordnets causes words such as *transfer certificates, TCs*, and *certificates* to be represented as different words, thereby resulting in incorrect grouping of queries in the FGG subsystem.

Poor quality emails and evolving queries affect the quality of FAQs generated by F-Gen. An example for evolving queries is as follows: Consider an e-learning system using a software namely *kibaco* to upload files. The query, *"How to upload files to kibaco"* is inappropriate when the institution replaces *kibaco* with a different software.

7. Conclusion

We present a novel F-Gen system for the automatic generation of FAQs from emails, which could be highly beneficial and cost-effective for large business enterprises. Answering pandemic-related emails is

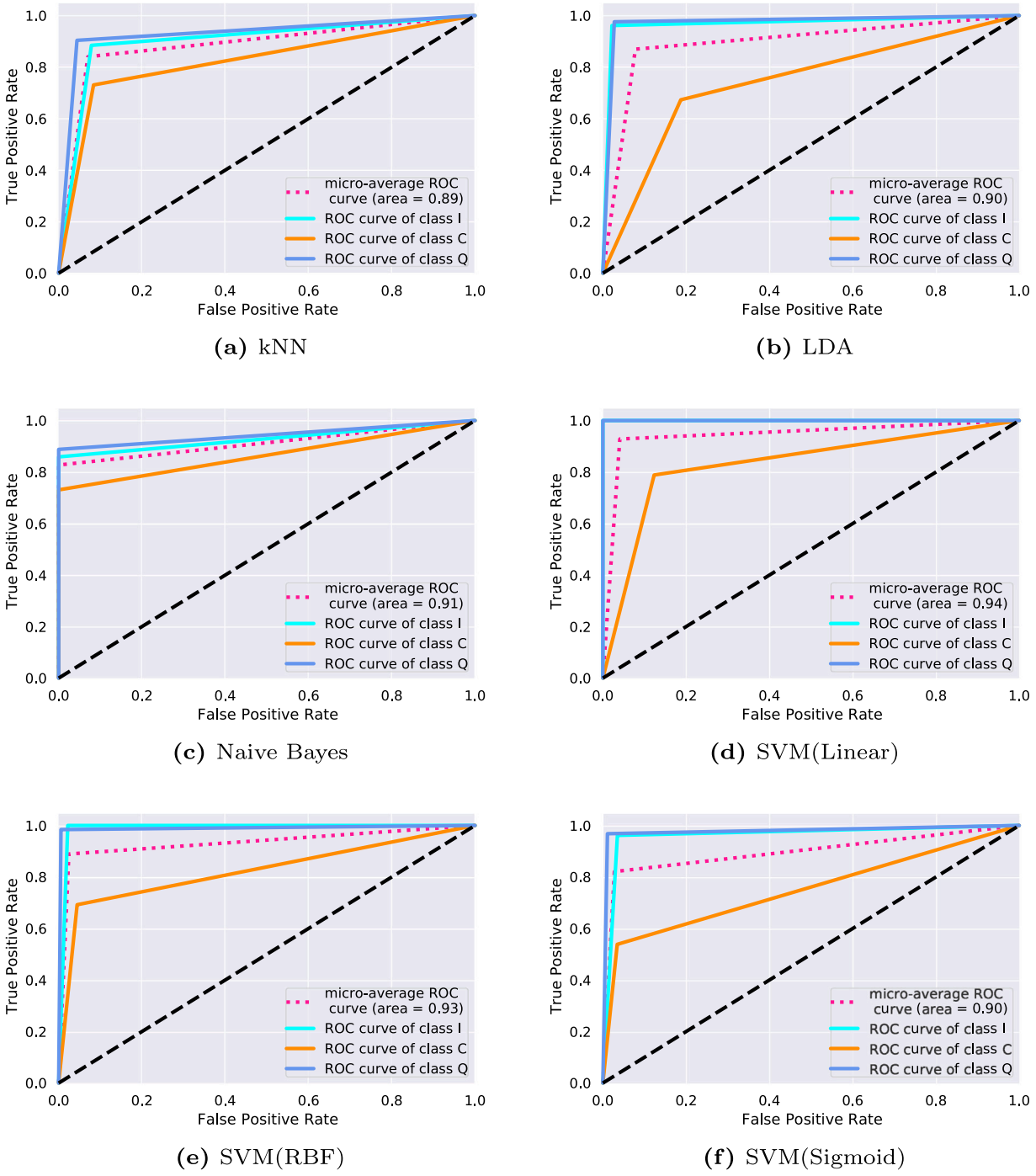

Fig. 10. ROC curves to estimate the best suited classifier for query classification.

Table 9
Possible errors and their causes.

S. No	Subsystem	Error type	Causes
1	QC	Misclassification	Email sentence overlaps across 2 or 3 classes
2	QC	Misclassification	Implicit queries
3	FGG	Incorrect formation of FAQ groups	Absence of domain specific wordnet
4	FG	Incomplete or Incorrect FAQs	Poor quality emails
5	FG	Inappropriate FAQs	Evolving queries

challenging (Matthewson, Tiplady, Gerakios, Foley, & Murphy, 2020), and F-Gen could be utilized to generate FAQs from them too. The specialty of the F-Gen system is its ability to implement different functionalities through various subsystems and finally integrate them into one system. F-Gen is unique in its design and implementation of all the components involved in the generation of valid FAQs from emails. Various deep learning methods (pre-trained word embeddings and Siamese LSTMs) were adopted for the subsystems. Experiments

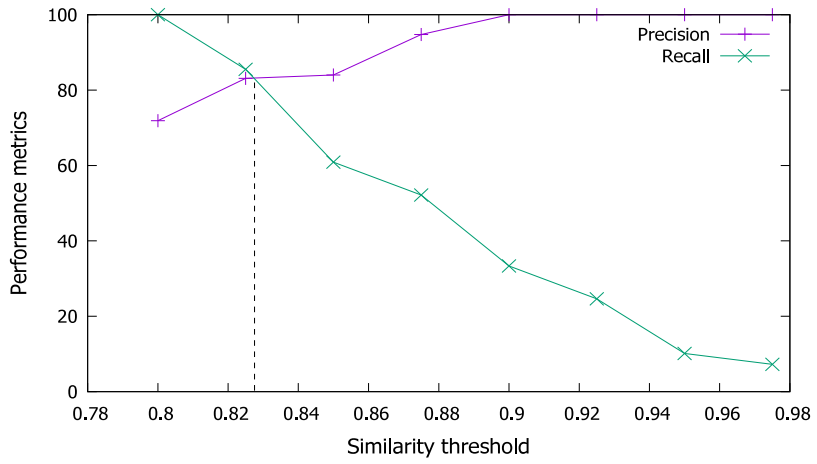

Fig. 11. Precision–Recall values against the similarity thresholds for query similarity classifier.

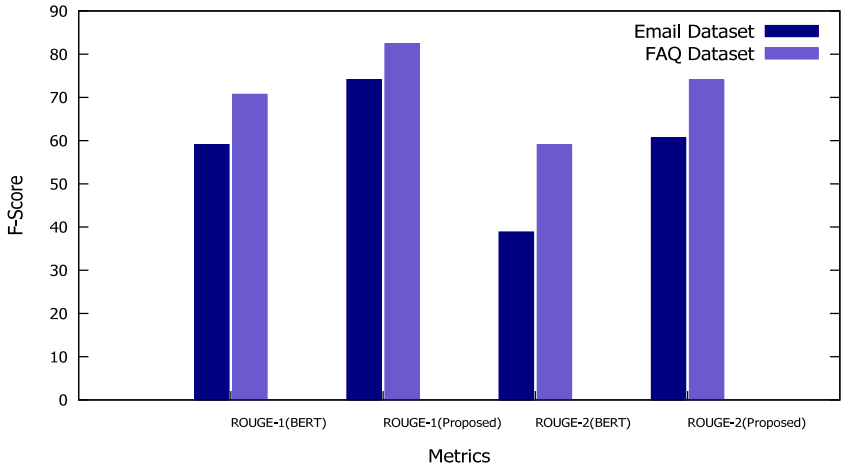

Fig. 12. F-Score of the proposed domain rich BERT summarization method vs. BERT summarization (Liu, 2019) for Email and FAQ datasets.

were conducted on a custom dataset specially crafted from a real-world email inbox as well as a public FAQ dataset. Some of the possible future works include FAQ generation from multilingual email inboxes, FAQ generation in a language different from the source (emails), computation of semantic similarity across email queries with varying contexts, distinguishing complex FAQs across FAQ lists, and handling a large number of candidate query pairs for FAQ generation given that there are mailboxes with millions of emails.

CRediT authorship contribution statement

Shiney Jeyaraj: Methodology, Software, Validation, Investigation, Data curation, Writing - original draft. **Raghuveera T.:** Supervision, Writing - review & editing, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by Anna University, Chennai, India through the Anna Centenary Research Fellowship.

References

Alami, N., Meknassi, M., & En-nahnahi, N. (2019). Enhancing unsupervised neural networks based text summarization with word embedding and ensemble learning. *Expert Systems with Applications*, *123*, 195–211.

Alibadi, Z., & Vidal, J. (2018). To read or to do? That's the task. In *Proceedings of the 2018 international conference on data science (ICDATA'18)* (pp. 279–285).

Altheneyan, A., & Menai, M. E. B. (2020). Evaluation of state-of-the-art paraphrase identification and its application to automatic plagiarism detection. *International Journal of Pattern Recognition and Artificial Intelligence*, *34*(04), Article 2053004.

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer normalization. arXiv:1607.06450.

Beck, D., Haffari, G., & Cohn, T. (2018). Graph-to-sequence learning using gated graph neural networks. arXiv:1806.09835.

Bozdogan, C., Zincir-Heywood, A. N., & Zincir, I. (2015). EMITS: An experience management system for IT management support. *International Journal of Software Engineering and Knowledge Engineering*, *25*(03), 513–550.

Cai, D., & Lam, W. (2020). Graph transformer for graph-to-sequence learning. In *Proceedings of the AAAI conference on artificial intelligence 5* (pp. 7464–7471).

Cohen, W. W., Carvalho, V. R., & Mitchell, T. M. (2004). Learning to classify email into "speech acts". In *Proceedings of the 2004 conference on empirical methods in natural language processing* (pp. 309–316).

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., et al. (2019). Unsupervised cross-lingual representation learning at scale. arXiv:1911.02116.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, *13*(1), 21–27.

Dataset (2017). https://www.kaggle.com/quora/question-pairs-dataset, last accessed 2021-04-20.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805.

Gillick, D., Brunk, C., Vinyals, O., & Subramanya, A. (2015). Multilingual language processing from bytes. arXiv:1512.00103.

Goyal, P., & Ferrara, E. (2018). Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, *151*, 78–94.
Guo, Z., Zhang, Y., Teng, Z., & Lu, W. (2019). Densely connected graph convolutional networks for graph-to-sequence learning. *Transactions of the Association for Computational Linguistics*, *7*, 297–312.
Heilman, M., & Smith, N. A. (2010). Good question! statistical ranking for question generation. In *The 2010 annual conference of the north american chapter of the association for computational linguistics* (pp. 609–617).
Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.
Ilie, G., & Ciocoiu, C. N. (2010). Application of fishbone diagram to determine the risk of an event with multiple causes. *Management Research and Practice*, *2*(1), 1–20.
Itakura, K., Kenmotsu, M., Oka, H., & Akiyoshi, M. (2010). An identification method of inquiry e-mails to the matching faq for automatic question answering. In *Distributed computing and artificial intelligence* (pp. 213–219). Berlin, Heidelberg: Springer.
Jeyaraj, S., & Tripuraribhatla, R. (2018). A framework for automatic generation of FAQs from email repositories. In *2018 IEEE 9th annual information technology, electronics and mobile communication conference (IEMCON, Vancouver* (pp. 1035–1041).
Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning Berlin, Heidelberg* (pp. 137–142).
Joshi, A., Fidalgo, E., Alegre, E., & Fernández-Robles, L. (2019). SummCoder: An unsupervised framework for extractive text summarization based on deep auto-encoders. *Expert Systems with Applications*, *129*, 200–215.
Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of tricks for efficient text classification. arXiv:1607.01759.
Kang, Y. B., Krishnaswamy, S., & Zaslavsky, A. (2013). A retrieval strategy for case-based reasoning using similarity and association knowledge. *IEEE Transactions on Cybernetics*, *44*(4), 473–487.
Kannan, A., Kurach, K., Ravi, S., Kaufmann, T., Tomkins, A., Miklos, B., et al. (2016). Smart reply: Automated response suggestion for email. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 955–964).
Kenter, T., Jones, L., & Hewlett, D. (2018). Byte-level machine reading across morphologically varied languages. In *Proceedings of the AAAI conference on artificial intelligence* (p. 1).
Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv:1412.6980.
Kolomiyets, O., & Moens, M. F. (2011). A survey on question answering technology from an information retrieval perspective. *Information Sciences*, *181*(24), 5412–5434.
Kusner, M., Sun, Y., Kolkin, N., & Weinberger, K. (2015). From word embeddings to document distances. In *International conference on machine learning* (pp. 957–966).
Kwong, H., & Yorke-Smith, N. (2009). Detection of imperative and declarative question-answer pairs in email conversation. In *Twenty-first international joint conference on artificial intelligence*.
Lample, G., & Conneau, A. (2019). Cross-lingual language model pretraining. arXiv:1901.07291.
Lee, J., Cho, K., & Hofmann, T. (2017). Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics*, *5*, 365–378.
Lee, D. D., & Seung, H. S. (2001). Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems* (pp. 556–562).
Lewis, D. D. (1998). Naive (Bayes) at forty: The independence assumption in information retrieval. In *European conference on machine learning, Berlin, Heidelberg* (pp. 4–15).
Liang, J., Cui, B., Jiang, H., Shen, Y., & Xie, Y. (2018). Sentence similarity computing based on word2vec and LSTM and its application in rice FAQ question-answering system. *Journal of Nanjing Agricultural University*, *41*(5), 946–953.
Lin, C. Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out* (pp. 74–81).
Liu, Y. (2019). Fine-tune BERT for extractive summarization. arXiv:1903.10318.
Matthewson, J., Tiplady, A., Gerakios, F., Foley, A., & Murphy, E. (2020). Implementation and analysis of a telephone support service during COVID-19. *Occupational Medicine*, *70*(5), 375–381.
Mihalcea, R., & Tarau, P. (2004). Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*.
Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).
Mueller, J., & Thyagarajan, A. (2016). Siamese recurrent architectures for learning sentence similarity. In *Thirtieth AAAI conference on artificial intelligence*.
Naeem, M. A., Linggawa, I. W. S., Mughal, A. A., Lutteroth, C., & Weber, G. (2018). A smart email client prototype for effective reuse of past replies. *IEEE Access*, *6*, 69453–69471.
Narayanan, A., Chandramohan, M., Venkatesan, R., Chen, L., Liu, Y., & Jaiswal, S. (2017). Graph2vec: Learning distributed representations of graphs. arXiv:1707.05005.
Nguyen, H. T., Duong, P. H., & Cambria, E. (2019). Learning short-text semantic similarity with word embeddings and external knowledge sources. *Knowledge-Based Systems*, *182*, Article 104842.
Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532–1543).
Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., et al. (2018). Deep contextualized word representations. arXiv:1802.05365.
Ramage, D., Hall, D., Nallapati, R., & Manning, C. D. (2009). Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 conference on empirical methods in natural language processing* (pp. 248–256).
Rath, M., Shah, C., & Floegel, D. (2017). Identifying the reasons contributing to question deletion in educational Q&A. In *Proceedings of the association for information science and technology* (pp. 327–336).
Ravi, S., Pang, B., Rastogi, V., & Kumar, R. (2014). Great question! question quality in community q&a. In *Eighth international AAAI conference on weblogs and social media*.
Report (2019). https://www.radicati.com/wp/wp-content/uploads/2018/12/Email-Statistics-Report-2019-2023-Executive-Summary.pdf, last accessed 2021-04-20.
Ruiz Costa-Jussà, M., Escolano Peinado, C., & Rodríguez Fonollosa, J. A. (2017). Byte-based neural machine translation. In *Proceedings of the first workshop on subword and character level models in NLP* (pp. 154–158). Association for Computational Linguistics.
Salminen, J., Hopf, M., Chowdhury, S. A., Jung, S. G., Almerekhi, H., & Jansen, B. J. (2020). Developing an online hate classifier for multiple social media platforms. *Human-Centric Computing and Information Sciences*, *10*(1), 1.
Sappelli, M., Pasi, G., Verberne, S., de Boer, M., & Kraaij, W. (2016). Assessing e-mail intent and tasks in e-mail messages. *Information Sciences*, *358*, 1–17.
Sarma, P. K., Liang, Y., & Sethares, W. A. (2018). Domain adapted word embeddings for improved sentiment classification. arXiv:1805.04576.
Singh, A., Ramasubramanian, K., & Shivam, S. (2019). Introduction to microsoft bot, RASA, and google dialogflow. In *Building an enterprise chatbot*. Berkeley, CA: A Press.
Sneiders, E., Sjöbergh, J., & Alfalahi, A. (2018). Automated email answering by text-pattern matching: Performance and error analysis. *Expert Systems*, *35*(1), Article e12251.
Sumikawa, Y., Fujiyoshi, M., Hatakeyama, H., & Nagai, M. (2019). An FAQ dataset for E-learning system used on a Japanese university. *Data in Brief*, *25*, Article 104001.
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).
Wang, D., Li, T., Zhu, S., & Gong, Y. (2010). iHelp: An intelligent online helpdesk system. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, *41*(1), 173–182.
Xu, K., Wu, L., Wang, Z., Feng, Y., Witbrock, M., & Sheinin, V. (2018). Graph2seq: Graph to sequence learning with attention-based neural networks. arXiv:1804.00823.
Xu, Z., & Yuan, H. (2020). Forum duplicate question detection by domain adaptive semantic matching. *IEEE Access*, *8*, 56029–56038.
Yang, X., Awadallah, A. H., Khabsa, M., Wang, W., & Wang, M. (2018). Characterizing and supporting question answering in human-to-human communication. In *41st international ACM SIGIR conference on research & development in information retrieval* (pp. 345–354).
Yelati, S., & Sangal, R. (2011). Novel approach for tagging of discourse segments in help-desk e-mails. In *2011 IEEE/WIC/ACM international conferences on web intelligence and intelligent agent technology* (pp. 369–372).
Zhang, Y., Lo, D., Xia, X., & Sun, J. L. (2015). Multi-factor duplicate question detection in stack overflow. *Journal of Computer Science and Technology*, *30*(5), 981–997.
Zhang, W. E., Sheng, Q. Z., Lau, J. H., Abebe, E., & Ruan, W. (2018). Duplicate detection in programming question answering communities. *ACM Transactions on Internet Technology (TOIT)*, *18*(3), 37.
Zhang, W. E., Sheng, Q. Z., Tang, Z., & Ruan, W. (2018). Related or duplicate: Distinguishing similar CQA questions via convolutional neural networks. In *41st international ACM SIGIR conference on research & development in information retrieval* (pp. 1153–1156).