Characterizing and Supporting Question Answering in Human-to-Human Communication

Xiao Yang*
Pennsylvania State University
xuy111@psu.edu

Ahmed Hassan Awadallah
Microsoft Research
hassanam@microsoft.com

Madian Khabsa*
Apple
madian@apple.com

Wei Wang
Microsoft Research
wawe@microsoft.com

Miaosen Wang*
Google
miaosen@google.com

ABSTRACT

Email is one of the most important means of online communication. People spend a significant amount of time sending, reading, searching and responding to email to manage tasks, exchange information, etc. In this paper, we focus on information exchange over enterprise email in the form of questions and answers. We study a large scale publicly available email dataset to characterize information exchange via questions and answers in enterprise email. We augment our analysis with a survey to gain insights on the types of questions exchanged, when and how do people get back to them and whether this behavior is adequately supported by existing email management and search functionality. We leverage this understanding to define the task of extracting question/answer pairs from threaded email conversations. We propose a neural network based approach that matches the question to the answer considering comparisons at different levels of granularity. We also show that we can improve the performance by leveraging external data of question and answer pairs. We test our approach using a manually labeled email data collected using a crowd-sourcing annotation study. Our findings have implications for designing email clients and intelligent agents that support question answering and information lookup in email.

KEYWORDS

Question Answering; Information Exchange in Email; Email Reply Assistance

ACM Reference Format:
Xiao Yang*, Ahmed Hassan Awadallah, Madian Khabsa, Wei Wang, and Miaosen Wang*. 2018. Characterizing and Supporting Question Answering in Human-to-Human Communication. In *SIGIR '18: 41st International ACM SIGIR Conference on Research and Development in Information Retrieval, July 8-12, 2018, Ann Arbor, MI, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3209978.3210046

*Work done while at Microsoft Research

SIGIR'18, July 8–12, 2018, Ann Arbor, MI, USA

ACM ISBN 978-1-4503-5657-2/18/07…$15.00
https://doi.org/10.1145/3209978.3210046

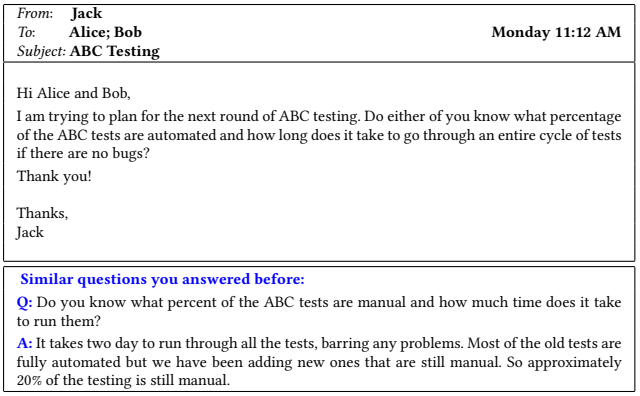

Figure 1: A motivational example showing how extracted question/answer pairs could be presented to users when a similar question is asked.

1 INTRODUCTION

Email is one of the most popular online activities and remains the major tool for communication and collaboration. In 2017, the total number of business and consumer emails sent and received per day was estimated to be 269 billion, and is expected to continue to grow at an average annual rate of 4.4% over the next four years, reaching 319.6 billion by the end of 2021 [1]. Email is particularly popular for work related communications with 86% of professionals naming email as their favorite mode of communication [1]. A recent survey shows that reading and answering emails takes up to 28% of enterprise workers' time, which is more than searching and gathering information (19%), communication and collaboration internally (14%), and second only to role specific tasks (39%) [11].

Dabbish et al. [16] developed a conceptual model of the main purpose email serves in an organizational context. They identified four distinct uses of email that have been previously studied in literature: task management, social communication, scheduling and information exchange. They conducted a survey with 124 participants to characterize different angles of email usage. They report that 36% of emails contained some information or attachment and 18% contained information requests. This shows that information exchange is one of the main uses of email. Email is often used to ask questions and respond to information requests. The ability to archive such information was shown to be one of the primary reasons users save messages [53, 54]. Previous research also shows that emails with information requests or responses are less likely

to be deleted by the user, and more likely to be left in the inbox or filed [16].

Previous research studied how people go back to information in their mailboxes. Some tend to create more structure in their mailboxes (especially with business email), where they organize their emails into categories and folders, while others rely on search to find emails [9]. We hypothesize that people are more likely to need to get back to email threads containing questions and answers and that providing adequate support for this category could assist users with retrieving this information or with sharing it with other people. Let's consider the example in Figure 1. Many users who have domain knowledge in specific areas tend to receive the same questions over and over (See Section 3 for details). The ability to extract question and answer pairs from threaded human-to-human communication could enable scenarios to assist users with composing responses to such messages. Additional experiences could allow users to directly ask questions and get answers if they think the answer is available in their mailboxes. This could also help unlock a lot of the organizational knowledge that would otherwise remain trapped inside people's mailboxes.

In this paper, we build on previous work by studying information exchange via question and answers in enterprise email. We present a detailed analysis of question and answer exchanges in a publicly available collection of almost 1,000,000 emails from a defunct information technology company. We complement the analysis with a survey of almost 1,000 information workers to gain more insights into the motivation for the observed email use. Motivated by the study of how people leverage email for information exchange, we propose a novel method for extracting question/answer pairs from email threads. We also perform a crowd-sourcing user annotation study to annotate pairs of question and candidate answers. We use this dataset to evaluate the proposed approach and show that we can efficiently identify question/answer pairs with reasonable performance outperforming multiple baselines. Finally, we study how we can leverage external question/answer datasets from community question answering forums to improve the overall performance of our approach.

Our contributions can be summarized as follows:

(1) We present detailed analysis of information exchange in enterprise email focusing on question and answer pairs in email thread.
(2) We conduct a survey with close to 1,000 information workers to gain insights on how they use email for information exchange. We study several aspects ranging from types of question and answers, when and how do people get back to this information, how long does the information remain relevant, etc.
(3) We define the task of extracting question and answer pairs from threaded email conversations. We present a novel approach for extracting such pairs and study the effect of leveraging external data to boost the performance. The proposed method outperforms multiple baseline methods.

The reminder of this paper will proceed as follows: We will start by discussing related work and contextually positioning the proposed work relative to the literature in Section 2. We present the analysis for characterizing information exchange in enterprise email in Section 3. We describe the datasets and the task definition in Section 4. Section 5 describes the method we propose for extracting question and answer pairs from threaded conversation and Section 6 describes our experiments and results. We conclude and discuss future work in Section 7.

2 RELATED WORK

Our work is related to several research areas, including email search and management, email intent understanding and community question answering. We cover each of them below:

2.1 Email Search and Management

Much of the early research on email focused on how people organized and managed their email. Whittaker and Sidner [54] proposed the concept of email overload to describe the usage of emails beyond communication needs, such as task management and personal archiving. They identified common strategies for handling email overload such as filing, searching and cleaning. Fisher et al. [20] found similar results in their study of mailboxes at a large tech company.

Grbovic et al. [21] show that, with the increase of email messages over time, users do not use folders and argue that search is an increasingly important alternative to human-generated folders and tags. Several studies have focused on developing effective search systems for email [19, 40]. Dumais et al. [19] found that email was the most commonly retrieved source of personal information (e.g. files, web history, emails, etc.). They also showed that people preferred to sort the results by date most often even when the default was best-match ranking. Craswell et al. [13] developed better ranking models for email search by combining email metadata with email content using BM25F. Ogilvie and Callan [36] proposed a language modeling approach to combine evidence from the text of the message, the subject, other messages in the thread, and messages that are in reply to the message.

Filing and adhoc search are some of the strategies people deploy to get back to email messages. In this work, we explore one source of information exchange (question/answer pairs) and provide analysis and methods for supporting people with getting back to this information in a more efficient way.

2.2 Email Intent Understanding

Previous research studied email acts and email intent analysis [3, 12, 28, 42]. Cohen et al. [12] proposed machine learning methods to classify emails according to an ontology of verbs and nouns, which describe the "email speech act" intended by the email sender. Follow-up work by Carvalho and Cohen [8] described a new text classification algorithm based on a dependency-network based collective classification method and showed significant improvements over a bag-of-words baseline classifier.

Another line of work studied the different actions people may perform against an email message. Dabbish et al. [16] examined people's ratings of message importance and the actions they took on specific email messages with a survey of 121 people. DiCastro et al. [17] studied four common user actions on email (read, reply, delete, delete-withoutread) using an opt-in sample of more than 100k users of the Yahoo! Mail service. They proposed and evaluated

a machine learning framework for predicting these four actions. Kooti et al. [27] characterized the replying behavior in conversations for pairs of users. They investigated the effects of increasing email overload on user behaviors and performed experiments on predicting reply time, reply length and whether the reply ends a conversion. Yang et al. [56] also studied reply behavior focusing on enterprise email. Recently, Lin et al. [30] proposed using a reparametrized recurrent neural network to model the actions that the recipient of the email might take upon receiving it.

Our work differs from the previous work in this area in several important ways. First, we focus on studying information exchange in enterprise communications. We characterize question and answer pairs exchanged over email and propose methods for extracting and linking them. Unlike previous work, we do not classify messages into intents or speech acts. We also do not try to predict actions such as reply, delete or forward. Our analysis is focused on question/answer exchange and goes beyond a single message to consider a threaded discussion with multiple messages.

2.3 Automated Response Generation

One of the areas that have seen increased attention recently is automatic response generation for emails or other media of conversation. Kannan et al. [25] proposed an end-to-end method for automatically generating short email responses. The system enabled the Smart Reply feature in Gmail. The features present users with a list of brief response candidates that she can select from to generate a complete email response. The paper describes a generative sequence-to-sequence model that is used to score a list of short candidate replies (less than 20 tokens). A follow-up paper described a more efficient method for solving the same problem [22]. Similar work has been done in the area of predicting response in natural language dialogues with domains like Twitter, movie dialogs, etc. For example, Ritter et al. [41] approached the problem from a machine translation perspective where a tweet is mapped to a response using phrase-based machine translation. Other recent works have also applied recurrent neural networks to full response prediction with applications to movie dialogs [43] or conversations from Weibo [45].

Our work is different from this line of work in that we do not attempt to generate a full response in a conversation, rather we focus on extracting question and answer pairs that already exist in previous emails. Full response generation in email targets frequent short responses where the responder intends to acknowledge, confirm, etc. a message with a short response. Our objective is different in multiple ways. First, we aim to extract personal knowledge in the form of question and answer pairs as opposed to frequent short responses. Second, while the question and answer pairs could be used for an experience that supports response generation, it could be also used for various other reasons such as search or creating archives of organizational knowledge in the form of questions and answers.

2.4 Community Question Answering

Another related research area is community question answering (CQA). In order to make use of the information and knowledge stored in a CQA archive, two important tasks have been formulated for questions: (1) retrieving related questions that already exist in the database [7, 24, 55]; and (2) finding the potentially desirable answers from the archived answers [4, 31, 46]. Depending on the techniques in use, prior work on CQA can be categorized into three groups: (1) statistical translation based methods; (2) latent variable based methods; and (3) deep learning based methods.

Early work frames this task as a translation problem: given two parallel corpora (questions and answers), a model learns the correlations from one corpus to another at word or phrase level. For example, [24] proposed to utilize IBM model 1 to learn a word translation probability matrix. The work of Cai et al. [7] and Zou et al. [59] generalized such method by also considering phrase alignments. Our method also incorporates translation models, however the final decision is made based on both the translation model and a matching-aggregating model described in Section 5. Latent variable models also attract much research attention. Under the assumption that relevant questions (or answers) should share a similar topic distribution, [7, 46] proposed first learning a topic model from the corpora, then computing the posterior probability that two sentences are drawn from a similar topic distribution. With the recent success of using vector representations of words in multiple natural language processing tasks, [29, 46, 57] incorporated pre-trained word embeddings in their latent variable models.

Recently, deep learning based models have led to rapid improvements for multiple natural language processing tasks [2, 6]. Therefore, researchers start to explore deep models for CQA tasks. Severyn and Moschitti [44] proposed to use deep convolutional neural networks to rank short text pairs. Santos et al. [18] combined deep convolutional neural networks and bag-of-word representations to model text similarity. Nakov et al. [34] proposed to predict the pairwise ranking of two candidate documents based on the learned sentence embeddings. In [58], two auto-encoders with shared intermediate representation are separately learned from two corpora. As a result, the need for parallel corpora is alleviated. However, such approaches only consider sentence level correlations, ignoring all word/phrase level overlapping or correlations. To address this issue, [32, 37, 51] proposed a matching-aggregating framework, where the comparison results between words are aggregated to make a final prediction. Our work follows this framework, however we consider comparisons from various levels of granularity, such as comparisons between words and phrases. We also employ a neural translation model to capture sentence-level context information.

3 CHARACTERIZING QUESTION ANSWER EXCHANGE IN EMAILS

3.1 Procedure

We conducted a survey to better understand how people exchange information via questions and answers in email. The survey was distributed to a random set of employees within a large technology company. 924 respondents completed the entire survey, while 13 additional respondents provided partial responses (response rate: 15%, completion rate: 99%). In our analysis, we only consider the 924 who completed the survey in its entirety. 72% of the respondents were male, and were distributed across a wide age range ranging from under 20 to more than 60. Respondents came from a diverse set of roles within the company including: software development,

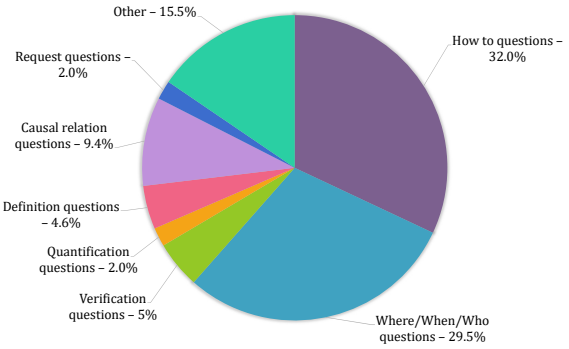

Figure 2: Question types distribution

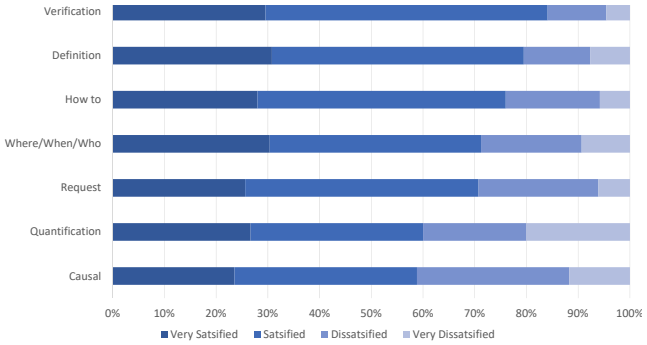

Figure 3: Question types satisfaction

program management, sales, marketing, legal, human resources, administrative assistance, IT support, finance, retail, etc.

The survey was structured into several sections, and was focused on work emails. In the first section, we asked our respondents about their general email behavior, such as the email client(s) that they use, and the number of emails that they typically receive during a work day.

In the second section, we asked our respondents to recall the last time they had a question and tried to locate the answer in their mailbox. This served the role of grounding the main content questions of the survey and helping the respondents to focus on a concrete situation. The following prompt was presented: *Recall the last time you were in the following position: You have a question in mind, and you know that the answer to that question exists in your work mailbox. So you try to get that answer from your email.*

Respondents were then asked several questions focusing on two main aspects. The first covered *what* information they were trying to locate, the type of question they had, whether they tend to need to get back to the same answer multiple time. The second focused on *how* they tried to locate this information and how successful they were at it.

3.2 Results

Regarding general email usage patterns, 98.7% of respondents indicated that they needed to locate answers from their mailbox to

Table 1: Types of questions used in the survey

Question Type	Examples
How to questions	How do I set up a new machine?
Where/When/Who questions	When is the next all-hands meeting?
Verification questions	Do we have a meeting today?
Quantification questions	How many items should we purchase?
Definition questions	What does X mean?
Causal relation questions	What are the reasons? What are the consequences?
Request questions	Can I have access to your computer?

questions they had at least once in the past month. 58.7% of respondents said this happened more than 10 times, 22.2% 5-10 times and 17.9% 1-5 times. Respondents also indicated that using the email search functionality is very popular with 78.9% using it more than once a day. We will pivot the results on three main aspects: what types of questions do people tend to get back to, how do they get back to the answers and how successful are they with that and how often do they need to get back to the same answer multiple times.

When asked to recall the last event they had a question with an answer in their mailbox, the majority of respondents (81.3%) were able to recall such an event. We asked the respondents to classify the question into a set of predefined categories based on the taxonomy presented in [39]. Table 1 shows the most frequently selected question types with an example for each type. Figure 2 shows the distribution of responses across question types. We note that the two most popular categories were how-to questions (32%) and when/where/who questions (29.5%). We also asked the respondents to describe the information they were looking for in free text. By analyzing the responses, we note that how-to questions are popular as people frequently need to locate instructions for performing certain tasks. When/Where/Who questions are popular as people try to find information about the time or location of events or the contact person they can ask about an event or a project. Other question types like verification, quantification, definition, causal relation and request questions were also present but with small proportions. Note that, given the way the questions were framed, this distribution does not necessarily reflect how often are such types of questions exchanged over email. Rather, it is biased toward the information that people are likely to need to get back to. For example, previous work [15] showed that requests are very popular in work emails. However, it is less likely for someone to try to get back to such questions compared to questions about instructions for doing a certain task or the contact person for some project.

We asked respondents about the strategies they employ for finding such information. 90.8% of the respondents indicated that they use the email search functionality. The remaining respondents used different strategies such as navigating to a folder or filtering by flagged messages and then browsing for the message. They also

indicated that they did not vividly remember the message that contains the answers in 36.1% of the times yet they knew they have answer in their mailbox. We also wanted to understand how well do these strategies support users in finding answers to previously asked questions in their mailboxes. We asked the respondents about their effort and success with respect to the process of locating the information. We considered them *Very Satisfied* if they located the information in less than one minute, *Satisfied* if they located the information in one to five minutes, *Dissatisfied* if they needed more than five minutes to locate the information and *Very Dissatisfied* if they failed to locate the information. We show the distribution of responses for each type of question in Figure 3. The figure shows the question types in order with easier questions (higher satisfaction) on the top. We note that Verification and Definition questions tend to be easier to locate while Causal and Quantification questions tend to be harder. The figure also highlights the opportunity to provide better support for locating this information which could increase the user efficiency by helping them locate answers faster and more successful.

Finally, we wanted to understand how repetitive is the behavior of locating answers to previously asked questions in email. We were particularity interested in understanding whether people tend to get back to the same answers multiple times or not. We started by considering how old is the message that the user found contained the answer. We found out that in 20% of the cases, the answer was received in a relatively recent message (within a week). 40% of the cases accounted for older messages (within a month), and the rest accounted for older messages. The respondents also indicated that they often needed to locate the answer to the same question multiple times. 44.2% of them reported that they have tried to locate the answer to their question before, while 45.4% indicated this was the first time they needed to do that. The rest of them could not recall. This behavior did not only happen with answers they have received but also with answers they previously shared to questions asked to them. 80% of the respondents indicated that they had to answer a question that they have already answered before via email. 46.8% of the respondents indicated that this happened 1-5 times within the last month, 21.4% indicated that it happened five to ten times and the rest mentioned that it happened more than ten times. The variance could be related to job roles with people working in roles such as IT support or customer support exhibiting this behavior more than other job roles.

4 DATA SETS AND TASK DEFINITION

The main task we pursue in this paper can be defined as follows: given email threads, identify all question answer pairs exchanged over this thread. This task can be further decomposed into two steps: (1) identify questions and (2) find the corresponding answers in the email threads. For questions identification, we apply simple rules to extract question sentences from Avocado research email collection [35], which are further filtered by crowd-sourcing annotations. For answer selection, we propose a novel model to select candidate answers to a given question. To train and evaluate our model, we collect an email question/answer dataset using a crowd-sourcing annotation platform. The collected dataset can serve as a benchmark and advance research study on this task.

We use the Avocado research email collection [35] from the Linguistic Data Consortium as our data source. This collection contains corporate emails from a defunct information technology company referred to as "Avocado". The collection contains the full content of emails, various meta information as well as attachments, folders, calendar entries, and contact details from Outlook mailboxes for 279 company employees. The full collection contains 938, 035 emails.

4.1 Question Identification and Analysis

We use a set of rules and human annotations to identify information seeking questions. First, all question sentences from the Avocado collection are extracted, such as those that start with one of six Ws (who, what, when, where, why and how) and end with question marks. Then, we filter out question sentences if the email they belong to is longer than a pre-defined threshold (e.g. 100 words). The rationale behind this is that we want to rule out the effects of other potentially irrelevant information that is contained in very long emails, and only focus on the question answering interaction. The question sentences are further filtered based on the crowd-sourced annotation results (described in Section 4.2) to remove invalid questions such as clauses and trivial questions such as "how are you?". Identifying information seeking questions is an interesting problem that has been studied before, e.g. [50]. We do not focus on proposing a solution to this problem in this work and rely on the rule-based approach followed by human annotation to extract questions with high precision. The extracted questions are later used to build our email question/answer dataset.

In order to understand what types of questions are frequently asked in the extracted sentences, we label each question sentence as type t if it contains a pre-defined keyword of the t-th question type. A list of the question types we considered is shown in Table 1. The keywords were provided in [39] and are generated based on several example questions for each type. For an instance, the keywords for the "Request" type questions include {can i, can we, shall we, ...}. Using these keywords, we are able to count the occurrence of different types of questions in the extracted large collection of sentences. Figure 4 shows the distribution of different question types. Comparing this distribution with Figure 2, we can see that we have much more "Request" and "Verification" questions in the extracted data. Recall that the survey was not asking the respondents about the questions they have asked or received over email. Rather, it was asking them about the questions and answers they needed to get back to later. Contrast this with the data analysis which characterizes the distribution of exchanged questions regardless of whether they have been needed again later or not. Note that the Avocado data does not include user interaction signals and hence information about which emails were visited later is not available. This difference in the distribution could indicate that while "Request" and "Verification" questions are common, it is less likely for users to need to look up their responses later. On the contrary, "How to" and "Where/When/Who" questions both occur with a relatively high frequency and are likely to be revisited.

We further carry out an analysis about the topics in the collected questions. While the analysis above focuses on the functionality of questions, here we investigate the content of these questions. In

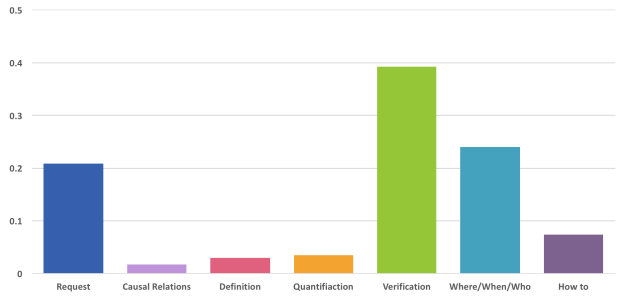

Figure 4: Distribution of different question types in the extracted question sentences from Avocado dataset.

Figure 5: Top words of certain topics visualized in the form of word cloud.

order to do that, we train a LDA [5] topic model on nouns and verbs of the extracted question sentences. For the number of pre-defined topics, we exhaustively tried 10, 20, 50, 100, 150 and empirically chose 50 which produced most interpretable results. Figure 5 shows the top words of four topics visualized in the form of word cloud. Based on the word cloud visualization, we can interpret that these four topics are mainly about people information, event and schedules, IT support and network connections, respectively. Question sentences that are labeled as one of these four topics by our LDA model constitute 25.6% of the entire questions. There were several other topics that includes daily life, travel information, etc. The topic modeling gives us more insights on the content of the questions in our dataset. It could also enable us to consider leveraging other datasets with similar topical distribution for pre-training our models.

4.2 Question/Answer Pairs Annotation

After the question identification step, we attempt to build an email question/answer dataset. We first locate each question sentence in its corresponding thread, then take the top 10 paragraphs from all subsequent emails as its candidate answers. The resulting <question, answer list> pairs are submitted to a crowd-sourcing platform to obtain annotations. During the crowd-sourcing annotation step, we also display the whole email body the question sentences belong to, in order to provide more information for annotators to understand the context.

Question	Agreement
(1) whether is a valid question	0.91
(2) whether is an enduring question	0.50
(3) select relevant answers	0.48

Table 2: Averaged inter-annotator agreements.

Each annotator is sequentially asked three inquiries for each <question, answer list> pair:

(1) select whether the question is a valid question;
(2) if answered "yes" to the first inquiry, then select whether the question is an *enduring* question;
(3) if answered "yes" to the above inquiries, then select the qualified answers from the candidate answers, or select "none of all above".

Since the simple rules that we used for question extraction may not be perfect, we again ask annotators to filter out non-question sentences. We also notice that many extracted questions are about trivial matters, such as "can you stop by my office now?", therefore we seek help from the annotators to only retain questions that convey valuable information which may be needed in the future.

Each <question, answer list> pair is annotated by three different annotators. The candidate answers with two or more votes is selected as a correct answer for each valid, enduring question. In total we obtained 1,695 unique questions. The averaged inter-annotator agreement for the three sequential inquiries are listed in Table 2. We achieved high inter-annotator agreement for the first inquiry, which is consistent with the way the question sentences are extracted. The second and third inquiries are more subjective, resulting in an inter-annotator agreement of 0.50 and 0.48, respectively.

5 METHOD

In this section, we describe our method for email question answering task. The proposed method consists of two components: (1) a matching-aggregating model which extends the decomposable attention model [37] by considering comparisons between words and phrases; and (2) a sequence-to-sequence translation based model to capture context information at sentence level.

5.1 Matching-Aggregating Model

Our matching-aggregating model extends the decomposable attention model [37] proposed by Parikh et al. Instead of only considering comparisons at word level, we also utilize comparisons between words and phrases. Since the semantic meaning of a text fragment may not be the simple combination of individual word semantics [48], using comparisons between words and phrases enables us to capture semantic meanings of phrases. In this way, we can explore more context information when making predictions. In the following part, we will first briefly describe the architecture of the decomposable attention model, then present our extension with word-phrase comparisons.

The decomposable attention model consists of three components, namely: (1) an attention module which soft-aligns words in one sentence to another. The attention weights are calculated based on the word embeddings; (2) a comparison module which compares

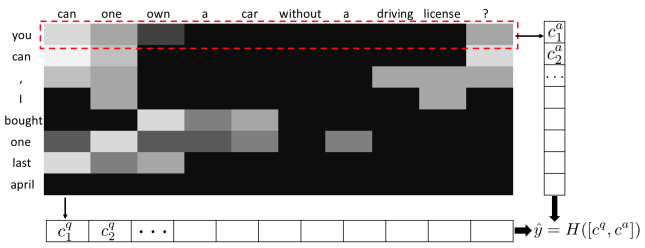

Figure 6: Illustration of the decomposable attention model.

each word in one sentence to its aligned sub-phrase in the other sentence; (3) an aggregation module which takes as input the summation of the comparison results and make the final prediction. A diagram of the decomposable attention model is illustrated in Figure 6.

Formally, if we denote a question sentence as $q = (q_1, q_2, \ldots, q_m)$, an answer sentence as $a = (a_1, a_2, \ldots, a_n)$, and the class label (relevant or irrelevant in this case) as y, then the objective is to maximize the probability $\Pr(y|q, a)$. Here we assume that each word $q_i, a_j \in \mathbb{R}^d$ is represented by d-dimensional word embedding vectors. In this work we use the pre-trained word embeddings from GloVe [38] to initialize each q_i and a_j.

Attention Module For each pair of words q_i and a_j, we can first calculate the (unnormalized) attention weights e_{ij} by:

$$e_{ij} = F(q_i)^T F(a_j) \tag{1}$$

with F being a non-linear transformation function. In other words, we first map the word embeddings into another space using F, then calculate the dot production as the attention weights. Afterwards, the normalized attention weights can be obtained using softmax functions along each sentence:

$$e^q_{ij} = \frac{\exp e_{ij}}{\sum_{k=1}^{m} \exp e_{kj}} \tag{2}$$

$$e^a_{ij} = \frac{\exp e_{ij}}{\sum_{k=1}^{n} \exp e_{ik}} \tag{3}$$

Finally, we use the weighted sum of individual word embeddings to represent the semantic meanings of the aligned sub-phrase:

$$\beta_i = \sum_{j=1}^{n} e^a_{ij} \cdot a_j \tag{4}$$

$$\alpha_j = \sum_{i=1}^{m} e^q_{ij} \cdot q_i \tag{5}$$

Comparison Module Now that we have the representations of each word and its aligned sub-phrase in the other sentence, we can compare them using another non-linear transformation function G:

$$c^q_i = G([q_i, \beta_i]) \tag{6}$$

$$c^a_j = G([a_j, \alpha_j]) \tag{7}$$

where the brackets denote concatenation.

Aggregation Module Based on the comparison results above, we can first aggregate them over each sentence by summation:

$$c^q = \sum_{i=1}^{m} c^q_i \tag{8}$$

$$c^a = \sum_{j=1}^{n} c^a_j \tag{9}$$

then concatenate the aggregated comparison results and feed it to a final classifier H:

$$\hat{y} = H([c^q, c^a]) \tag{10}$$

In practice, F, G, H are implemented by two-layer feed-forward networks with ReLU non-linearity.

The original decomposable attention model determines attention weights sorely based on individual word embeddings, therefore unable to capture context information in phrases during attention weights calculation. Such attention mechanism is sometimes referred to as *individual attentions* [14]. In order to overcome this problem, we propose extending decomposable attention model by considering comparisons at different levels of granularities. The architecture is illustrated in Figure 7. First, a convolutional neural network (CNN) is employed to learn a hierarchy of representations for each sentence. These representations capture semantics of phrases with different lengths. Afterwards, the learned representations are used in addition to word embeddings to calculate attention weights. More specifically, for a sentence $q = (q_1, q_2, \ldots, q_m)$, we first feed it to a convolutional neural network to obtain a hierarchy of feature representations:

$$\{q^{(1)}, q^{(2)}, \ldots\} = \text{CNN}(q) \tag{11}$$

where $q^{(t)}$ denotes the feature maps after the t-th convolutional block. A convolutional block consists of two convolutional layers, each followed by a Rectifier Linear Unit (ReLU) layer. If after the first convolutional block the receptive field is 5, then $q^{(1)}$ models the semantics of all 5-grams. In this way, features at higher levels gradually encode the semantics of longer phrases. A similar process can be applied to the answer sentence a. To unify the notation, we can define:

$$q^{(0)} = q \tag{12}$$

$$a^{(0)} = a \tag{13}$$

The original decomposable attention model can be seen as a score function between a pair of sentence representations. With the learned hierarchical representations, we can apply decomposable attention model to different pairs of representations. For example, we can pair a with each $q^{(t)}$ and feed them to the decomposable attention model. Similarly, we can pair q and each $a^{(\tau)}$. If three levels of features are learned in our convolutional neural network for both questions and answers, then in total we need to score five pairs of sentence representations.

5.2 Translation-based Language Model

We complement the matching-aggregating model with a translation-based language model to capture sentence-level information. Previous translation models [24] attempted to solve this problem based

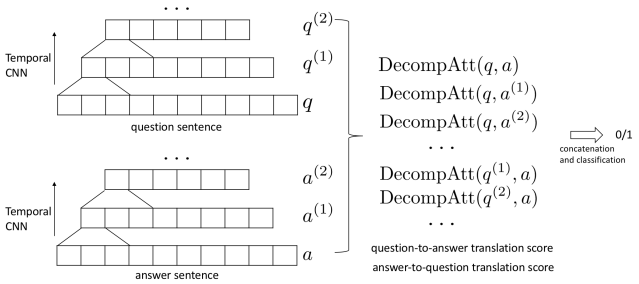

Figure 7: Architecture of the proposed model. DecompAtt stands for a decomposable attention model that is applied to a pair of representations.

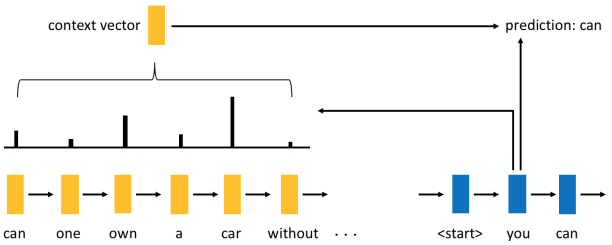

Figure 8: Illustration of the attention-based sequence-to-sequence model.

on word translation probability matrix. In this work, we utilize sequence-to-sequence models to model question-to-answer and answer-to-question translations. In this way, we are able to better understand context information, especially for long sentences. Given a question/answer pair, our model sequentially scores every words in question (or answer) sentence conditioned on previous words and the answer (or question) sentence. The resulting scores are concatenated together with the learned representations to make a final prediction.

Our sentence-level translation models are implemented by attention-based sequence-to-sequence models [26, 49]. The architecture is illustrated in Figure 8. During training, two separate models are learned to model question-to-answer and answer-to-question translations, in order to capture context information in both questions and answers. Each model consists of a 1-layer Long Short-Term Memory (LSTM) [23] as an encoder and a 1-layer LSTM as an decoder. The hidden state size is 128.

6 EXPERIMENTS

In this section we evaluate our method on the proposed email question/answer dataset. We also compare with several baseline methods on this dataset.

6.1 Dataset and Evaluation

Email Question Answering Dataset The proposed dataset (referred to as Avocado QA) contains 1,695 unique questions, each associated with 10 candidate answers. The questions and answers are extracted from Avocado research email collection [35]. One or more answers are labeled as relevant by annotators while the others are labeled as irrelevant.

Community Question Answering Dataset We use SemEval 2017 Task 3 dataset [33] which is collected from Qatar Living website. This dataset contains 317 original questions, each associated with 100 answers (comments). Every answer is labeled as either relevant or irrelevant to its corresponding question. This dataset is referred to as SemEval 2017. We use this data to study whether pre-training on an out-of-domain dataset will improve performance.

Evaluation For each method, we report its mean average precision (MAP) and average recall (AvgRec) evaluated at top 10 retrieved results. These evaluation metrics are widely used for information retrieval tasks.In parallel to terminologies used in information retrieval, for each query we can calculate the average precision and average recall from the top K ranked results. Average precision is then averaged over all queries to give mean average precision (MAP):

$$\text{MAP} = \frac{1}{Q}\sum_{q=1}^{Q}\frac{1}{m}\sum_{\substack{k=1 \\ D_k \text{is relevant}}}^{K}\text{Precision}(D^q_{1:k}) \tag{14}$$

where Q is the total number of queries, $D^q_{1:k}$ represents the top k ranked results for query q, and m is the total number of relevant documents within top K results. Similarly, average recall (AvgRec) can be defined as:

$$\text{AvgRec} = \frac{1}{Q}\sum_{q=1}^{Q}\frac{1}{m}\sum_{\substack{k=1 \\ D_k \text{is relevant}}}^{K}\text{Recall}(D^q_{1:k}) \tag{15}$$

6.2 Baseline Methods

In this section we describe several representative baseline methods that are evaluated in our experiments.

TFIDF is a frequently used method for information retrieval tasks. It represents each document by a vector representation and ranks the documents based on their cosine similarity to queries. Each element in the vector representation is weighted by the the multiplication of term frequency and inverse document frequency [47].

Bilateral Multi-Perspective Matching Model (BiMPM) is first proposed by Wang et al. [52] for text matching and achieved state-of-the-art results on several natural language processing tasks including question answering and natural language inference. This model first learns representations using a recurrent neural network, then fuses four different matching strategies (full matching, max-pooling matching, attentive matching and max-attentive matching) to compare words from two sentences. Finally the comparison results are aggregated to make a prediction.

Decomposable Attention Model (DecompAtt) Proposed by Parikh et al. [37], the decomposable attention model follows the matching-aggregating framework for text matching. However, it eliminates the use of any recurrent neural networks or convolutional neural networks, and is solely based on attention mechanisms. It first aligns each word in one sentence with a sub-phrase in another sentence, then exhaustively compares all such aligned pairs to make final predictions. This method can be seen as a direct baseline for our model, in order to justify the effectiveness of our modifications.

Method	MAP	AvgRec
TFIDF	32.89	71.18
BiMPM	42.05	75.64
DecompAtt	45.29	79.79
Ours	**48.95**	**82.34**

Table 3: Results of different methods when training and testing on Avocado QA dataset.

Method	MAP	AvgRec
Ours (without pre-training)	48.95	82.34
Ours (with pre-training)	**50.75**	**82.55**

Table 4: Results of our method with and without pre-training on SemEval 2017 dataset.

6.3 Results on Avocado QA Dataset

In this section we show the results of different methods on the proposed Avocado QA dataset. For each method in the experiments, we adopt 10-fold cross validation and report the mean performance. The results are summarized in Table 3. The TFIDF baseline achieves a MAP of 32.89, much lower than other machine learning based methods. Such results demonstrate the "lexical gap" [10] issue of question answering, where words in questions may not necessarily occur in answers. All embedding based methods (BiMPM, DecompAtt, Ours) outperform TFIDF with a large margin, showing the superiority of using deep representations. As a direct comparison, our method achieves better performance compared with the original decomposable attention model under all evaluation metrics. Such results demonstrate the advantages of considering comparisons between words and phrases and incorporating a neural translation model to capture sequence-level information. Our modifications to the matching-aggregating framework can be beneficial in general for many NLP tasks such as natural language inference and sentiment analysis. The BiMPM model achieves the worst results among embedding based methods. Since it contained more parameters to be learned, we hypothesize that more data are needed for effectively training the BiMPM model.

6.4 Results on Avocado QA Dataset with Pre-training

In this section, we investigate whether pre-training on an out-of-domain dataset will improve performance. Based on our topic modeling analysis in Section 4.1, we choose to use SemEval 2017 dataset for pre-training since it covers similar topics such as IT support, events, daily life and travel information. We use all training data from SemEval 2017 for pre-training, and adopt a 10-fold cross validation when fine-tuning and testing on the Avocado QA dataset. The mean performances are reported in Table 4. We can observe that the model with pre-training outperforms the one without pre-training: 1.80 and 0.21 improvements for MAP and AvgRec, respectively. Such improvements suggest that an out-of-domain dataset can be utilized for pre-training for email question answering task, even if the topics of two datasets do not perfectly overlap.

7 CONCLUSIONS

In this paper, we investigate information exchange over enterprise emails in the form of questions and answers. We study a large scale publicly available email dataset (Avocado) to characterize the question taxonomies and topics of question/answer pairs. The study is further augmented with a survey to gain insights on the types of questions exchanged, when and how do people get back to them and whether this behavior is adequately supported by existing email management and search functionality. We found out that answers to certain types of questions (about instructions, events, people, etc.) are morel likely to be needed again. And while most people use the email search functionality to locate this information, many of them end up failing or spending a long time to locate the answers. We leverage this understanding to define the task of extracting question/answer pairs from threaded email conversations. To accomplish this task, we propose a neural network based approach that matches the answer to the question using a matching-aggregating model that considers comparisons between words and phrases; and a sequence-to-sequence model to capture context information at sentence level. We show that our model can outperform state-of-the art baselines and that the overall performance can be further improved by leveraging external data of question and answer pairs for pre-training. To evaluate the proposed approach, we collect an email question answering dataset from Avocado using a crowd-sourcing annotation platform. Our findings and approaches have implications on designing new intelligent assistant scenarios to support question answering and information lookup in email.

REFERENCES

[1] 2015. Email Statistics Report. The Radicati Group, INC.. (2015). https://goo.gl/brmqrm

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[3] Paul N. Bennett and Jaime Carbonell. 2005. Detecting Action-items in e-Mail. In *SIGIR '05*. ACM, New York, NY, USA, 585–586.

[4] Adam Berger, Rich Caruana, David Cohn, Dayne Freitag, and Vibhu Mittal. 2000. Bridging the lexical chasm: statistical approaches to answer-finding. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 192–199.

[5] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[6] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326* (2015).

[7] Li Cai, Guangyou Zhou, Kang Liu, and Jun Zhao. 2011. Learning the Latent Topics for Question Retrieval in Community QA.. In *IJCNLP*, Vol. 11. 273–281.

[8] Vitor R. Carvalho and William W. Cohen. 2005. On the Collective Classification of Email "Speech Acts". In *SIGIR '05*. ACM, New York, NY, USA, 345–352.

[9] M.E. Cecchinato, A. Sellen, M. Shokouhi, and G. Smyth. 2016. Finding email in a multi-account, multi-device world. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. 1200–1210. https://doi.org/10.1145/2858036.2858473

[10] Long Chen, Joemon M Jose, Haitao Yu, Fajie Yuan, and Dell Zhang. 2016. A semantic graph based topic model for question retrieval in community question answering. In *WSDM*. ACM, 287–296.

[11] Michael Chui, James Manyika, Jacques Bughin, Richard Dobbs, Charles Roxburgh, Hugo Sarrazin, Georey Sands, and Magdalena Westergren. 2012. The social economy: Unlocking value and productivity through social technologies. McKinsey Global Institute.. (2012).

[12] William W. Cohen, Vitor R. Carvalho, and Tom M. Mitchell. 2004. Learning to classify email into speech acts. In *In Proceedings of Empirical Methods in Natural Language Processing*.

[13] Nick Craswell, Hugo Zaragoza, and Stephen Robertson. 2005. Microsoft Cambridge at TREC 14: Enterprise Track. In *Proceedings of the Fourteenth Text REtrieval Conference, TREC 2005, Gaithersburg, Maryland, USA, November 15-18,*

2005. http://trec.nist.gov/pubs/trec14/papers/microsoft-cambridge.enterprise.pdf
[14] Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. 2016. Attention-over-attention neural networks for reading comprehension. *arXiv preprint arXiv:1607.04423* (2016).
[15] L. A. Dabbish and R. E. Kraut. 2006. Email overload at work: An analysis of factors associated with email strain. In *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work (CSCW '06)*. 431–440. https://doi.org/10.1145/1180875.1180941
[16] Laura A. Dabbish, Robert E. Kraut, Susan Fussell, and Sara Kiesler. 2005. Understanding Email Use: Predicting Action on a Message. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '05)*. ACM, New York, NY, USA, 691–700.
[17] Dotan Di Castro, Zohar Karnin, Liane Lewin-Eytan, and Yoelle Maarek. 2016. You'Ve Got Mail, and Here is What You Could Do With It!: Analyzing and Predicting Actions on Email Messages. In *WSDM '16*. 307–316.
[18] Cicero Dos Santos, Luciano Barbosa, Dasha Bogdanova, and Bianca Zadrozny. 2015. Learning hybrid representations to retrieve semantically equivalent questions. In *ACL-IJCNLP*, Vol. 2. 694–699.
[19] S. Dumais, E. Cutrell, J. J. Cadiz, G. Jancke, R. Sarin, and D. C. Robbins. 2003. Stuff I've seen: A system for personal information retrieval and re-use. In *ACM SIGIR Forum*, Vol. 49. ACM, 28–35.
[20] D. Fisher, A. J. Brush, E. Gleave, and M. Smith. 2006. Revisiting Whittaker & Sidnerś "email overload" ten years later. In *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work (CSCW '06)*. Banff, Alberta, Canada, 309–312. https://doi.org/10.1145/1180875.1180922
[21] M. Grbovic, G. Halawi, Z. Karnin, and Y. Maarek. 2014. How many folders do you really need?: Classifying email into a handful of categories. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (CIKM '14)*. Shanghai, China, 869–878. https://doi.org/10.1145/2661829.2662018
[22] Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient Natural Language Response Suggestion for Smart Reply. *CoRR* abs/1705.00652 (2017). arXiv:1705.00652 http://arxiv.org/abs/1705.00652
[23] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
[24] Jiwoon Jeon, W Bruce Croft, and Joon Ho Lee. 2005. Finding similar questions in large question and answer archives. In *Proceedings of the 14th ACM international conference on Information and knowledge management*. ACM, 84–90.
[25] Anjuli Kannan, Karol Kurach, Sujith Ravi, Tobias Kaufmann, Andrew Tomkins, Balint Miklos, Greg Corrado, László Lukács, Marina Ganea, Peter Young, et al. 2016. Smart reply: Automated response suggestion for email. *arXiv preprint arXiv:1606.04870* (2016).
[26] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. *arXiv preprint arXiv:1701.02810* (2017).
[27] Farshad Kooti, Luca Maria Aiello, Mihajlo Grbovic, Kristina Lerman, and Amin Mantrach. 2015. Evolution of Conversations in the Age of Email Overload. In *WWW '15*. ACM, 603–613.
[28] Andrew Lampert, Robert Dale, and Cecile Paris. 2010. Detecting Emails Containing Requests for Action. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT '10)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 984–992.
[29] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *ICML*. 1188–1196.
[30] Chu-Cheng Lin, Dongyeop Kang, Michael Gamon, Madian Khabsa, Ahmed Hassan Awadallah, and Patrick Pantel. 2017. Actionable Email Intent Modeling with Reparametrized RNNs. *arXiv preprint arXiv:1712.09185* (2017).
[31] Zhengdong Lu and Hang Li. 2013. A deep architecture for matching short texts. In *NIPS*. 1367–1375.
[32] Mitra Mohtarami, Yonatan Belinkov, Wei-Ning Hsu, Yu Zhang, Tao Lei, Kfir Bar, Scott Cyphers, and Jim Glass. 2016. SLS at SemEval-2016 Task 3: Neural-based approaches for ranking in community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. 828–835.
[33] Preslav Nakov, Doris Hoogeveen, Lluís Màrquez, Alessandro Moschitti, Hamdy Mubarak, Timothy Baldwin, and Karin Verspoor. 2017. SemEval-2017 task 3: Community question answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. 27–48.
[34] Preslav Nakov, Lluís Màrquez, and Francisco Guzmán. 2016. It Takes Three to Tango: Triangulation Approach to Answer Ranking in Community Question Answering. In *EMNLP*. 1586–1597.
[35] Douglas Oard, William Webber, David Kirsch, and Sergey Golitsynskiy. 2015. Avocado Research Email Collection. DVD. (2015).
[36] P. Ogilvie and J. Callan. 2005. Experiments with language models for known-item finding of e-mail messages. In *TREC*.
[37] Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933* (2016).
[38] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*. 1532–1543.
[39] Jeffrey Pomerantz. 2005. A linguistic analysis of question taxonomies. *Journal of the Association for Information Science and Technology* 56, 7 (2005), 715–728.
[40] Pranav Ramarao, Suresh Iyengar, Pushkar Chitnis, Raghavendra Udupa, and Balasubramanyan Ashok. 2016. InLook: Revisiting Email Search Experience. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '16)*. ACM, New York, NY, USA, 1117–1120. https://doi.org/10.1145/2911451.2911458
[41] Alan Ritter, Colin Cherry, and William B. Dolan. 2011. Data-driven Response Generation in Social Media. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Stroudsburg, PA, USA, 583–593.
[42] Maya Sappelli, Gabriella Pasi, Suzan Verberne, Maaike de Boer, and Wessel Kraaij. 2016. Assessing e-mail intent and tasks in e-mail messages. *Inf. Sci.* 358-359 (2016), 1–17.
[43] Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. 2015. Hierarchical Neural Network Generative Models for Movie Dialogues. *CoRR* abs/1507.04808 (2015). arXiv:1507.04808 http://arxiv.org/abs/1507.04808
[44] Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*.
[45] Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural Responding Machine for Short-Text Conversation. In *Proceedings of the 53th Annual Meeting of the Association of Computational Linguistics*.
[46] Yikang Shen, Wenge Rong, Zhiwei Sun, Yuanxin Ouyang, and Zhang Xiong. 2015. Question/Answer Matching for CQA System via Combining Lexical and Sequential Information.. In *AAAI*. 275–281.
[47] Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation* 28, 1 (1972), 11–21.
[48] Michael Stubbs. 2001. *Words and phrases: Corpus studies of lexical semantics*. Blackwell publishers Oxford.
[49] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*. 3104–3112.
[50] Kai Wang and Tat-Seng Chua. 2010. Exploiting Salient Patterns for Question Detection and Question Retrieval in Community-based Question Answering. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING '10)*. 1155–1163.
[51] Shuohang Wang and Jing Jiang. 2016. A Compare-Aggregate Model for Matching Text Sequences. *arXiv preprint arXiv:1611.01747* (2016).
[52] Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. *arXiv preprint arXiv:1702.03814* (2017).
[53] S. Whittaker, T. Matthews, J. Cerruti, H. Badenes, and J. Tang. 2011. Am I wasting my time organizing email?: A study of email refinding. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Vancouver, BC, Canada, 3449–3458. https://doi.org/10.1145/1978942.1979457
[54] S. Whittaker and C. Sidner. 1996. Email overload: Exploring personal information management of email. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '96)*. Vancouver, British Columbia, Canada, 276–283. https://doi.org/10.1145/238386.238530
[55] Xiaobing Xue, Jiwoon Jeon, and W Bruce Croft. 2008. Retrieval models for question and answer archives. In *Proceedings of the 31st annual international ACM SIGIR*. ACM, 475–482.
[56] Liu Yang, Susan T. Dumais, Paul N. Bennett, and Ahmed Hassan Awadallah. 2017. Characterizing and Predicting Enterprise Email Reply Behavior. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '17)*. ACM, New York, NY, USA, 235–244.
[57] Guangyou Zhou, Tingting He, Jun Zhao, and Po Hu. 2015. Learning Continuous Word Embedding with Metadata for Question Retrieval in Community Question Answering.. In *ACL*. 250–259.
[58] Guangyou Zhou, Yin Zhou, Tingting He, and Wensheng Wu. 2016. Learning semantic representation with neural networks for community question answering retrieval. *Knowledge-Based Systems* 93 (2016), 75–83.
[59] Will Y Zou, Richard Socher, Daniel Cer, and Christopher D Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *EMNLP*. 1393–1398.