Context-Aware Intent Identification in Email Conversations

Wei Wang, Saghar Hosseini, Ahmed Hassan Awadallah, Paul N. Bennett and Chris Quirk
Microsoft
wawe,sahoss,hassanam,pauben,chrisq@microsoft.com

ABSTRACT

Email continues to be one of the most important means of online communication. People spend a significant amount of time sending, reading, searching and responding to email in order to manage tasks, exchange information, etc. In this paper, we study intent identification in workplace email. We use a large scale publicly available email dataset to characterize intents in enterprise email and propose methods for improving intent identification in email conversations. Previous work focused on classifying email messages into broad topical categories or detecting sentences that contain action items or follow certain speech acts. In this work, we focus on sentence-level intent identification and study how incorporating more context (such as the full message body and other metadata) could improve the performance of the intent identification models. We experiment with several models for leveraging context including both classical machine learning and deep learning approaches. We show that modeling the interaction between sentence and context can significantly improve the performance.

CCS CONCEPTS

• **Information systems** → **Email**; **Clustering and classification**; • **Computing methodologies** → **Supervised learning by classification**; **Neural networks**;

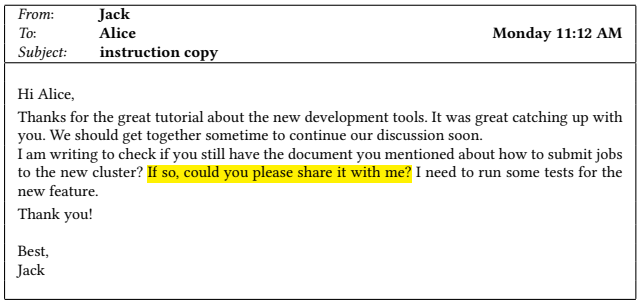
KEYWORDS

Email Intent Understanding, Actionable Intents, Context Augmented Classification

ACM Reference Format:
Wei Wang, Saghar Hosseini, Ahmed Hassan Awadallah, Paul N. Bennett and Chris Quirk. 2019. Context-Aware Intent Identification in Email Conversations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19), July 21–25, 2019, Paris, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3331184.3331260

1 INTRODUCTION

Email is one of the most popular online activities and remains a major tool for communication and collaboration. In 2017, it is estimated that 269 billion emails were sent and received per day and that the total volume of emails is expected to continue to grow reaching 319.6 billion by the end of 2021 [1]. Email is particularly popular for work related communications. 86% of professionals named email as their favorite mode of communication [1]. Enterprise workers tend to spend a lot of time on email too. 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%) and communication and collaboration internally (14%). [7].


From: **Jack**
To: **Alice** **Monday 11:12 AM**
Subject: **instruction copy**

Hi Alice,

Thanks for the great tutorial about the new development tools. It was great catching up with you. We should get together sometime to continue our discussion soon.
I am writing to check if you still have the document you mentioned about how to submit jobs to the new cluster? If so, could you please share it with me? I need to run some tests for the new feature.

Thank you!

Best,
Jack

Figure 1: Example of an email where the sender is *requesting information* from the recipient. The request sentence is highlighted in yellow

Dabbish et al. [14] developed a conceptual model of the main purpose email serves in an organizational context. They conducted a survey of 124 participants to characterize different aspects of email usage. Based on this, they identified four distinct uses of email that have been previously studied in literature: task management, social communication, scheduling, and information exchange. Previous work also studied how to detect emails that have requests for actions [24] and showed that such emails are less likely to be deleted by the user, and more likely to be left in the inbox or filed [14]. Studying and detecting intent in enterprise communications can enable us to better understand how information workers use email and how we can integrate machine intelligence into email systems and build smart email clients to provide more value for email users. For instance, understanding intents like intent to *set up a meeting* could enable us to create new intelligent experiences that can offer to assist the user with scheduling the meeting. Understanding that a user is making a *commitment* or a *promise* to perform a certain action, allows the system to help users track tasks in their to-do list. Such actions could be recommended to users and performed on their behalf upon confirmation. They could be surfaced in email clients or offered by a digital assistant.

In this paper, we build on previous work by studying intents in email conversations. We follow the footsteps of Dabbish et al. [14] by studying the different uses of email. Instead of using survey-based methods, we conduct large scale analysis of a publicly available enterprise email dataset: the *Avocado* corpus [27]. We leverage this analysis to study the problem of detecting intents in email conversations. Previous work, e.g., [24] and [4], has focused on sentence-level intent detection by detecting the sentence(s) which express the target intents in an email conversation. This setup

SIGIR '19, July 21–25, 2019, Paris, France

ACM ISBN 978-1-4503-6172-9/19/07…$15.00
https://doi.org/10.1145/3331184.3331260

is useful because it allows downstream applications to leverage the sentences where the action item was mentioned. For instance, a TO-DO application could simply transform the sentences with *commitment* intent to tasks in the to-do list.

This work also focuses on the task of sentence-level intent identification and extends previous work [4, 24] in two ways. First, we focus on detecting intent at a finer grain (e.g. request for setting up a meeting, sharing a document, etc.) as opposed to the generic notion of an *action item*. Second, we study how we can leverage context to improve sentence-level email intent detection. We use the term *context* to refer to additional information outside of the target sentence such as the text of the email message before and after the target sentences. Previous work mostly used linguistic features from the sentence itself [10], and ignored a rich set of contextual information available in the rest of the message. We hypothesize that the sentence-level intent identification could benefit from the complementary information available in the context. Consider the example in Figure 1. The highlighted sentence represents a sentence where the sender is requesting information from the recipient. To recognize the intent behind this sentence, it is important to consider the context provided in the rest of the message. The range of speech acts that may be indicative in context of an intent are often not obvious and may be influenced by both social norms and the intent being modeled. For example, a sentence near the beginning of a mail demonstrating gratitude may indicate a request is likely to occur later in the mail. Furthermore, typically only the target intent (e.g. "requesting information") is labeled. Hence, we study the effect of context information for sentence-level user intent identification in email conversations and explore different ways of leveraging that information.

Our contributions can be summarized as follows:

(1) We present detailed analysis of email usage in enterprise settings using a large scale publicly available email collection.
(2) We study the problem of incorporating context information for identifying intents in email messages and show that humans benefit from contextual information when identifying intents in emails.
(3) We propose several methods for incorporating context in sentence-level intent identification and show that incorporating context significantly improves performance.

The remainder of this paper will proceed as follows: In Section 2, we discuss related work and position our work with respect to the literature. We present an analysis of email usage in enterprise settings using a large scale publicly available email collection in Section 3. Section 4 describes the method we propose for leveraging context information for identifying intents in email messages and Section 5 describes our experiments and results. We conclude and discuss future work in Section 6.

2 RELATED WORK

Our work is related to several lines of work, including email search and management, email intent understanding, and email classification in general. 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 [34] 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. Grbovic et al. [19] showed 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 [16, 31]. Others focused on developing better ranking models for email search [12, 33]. Craswell et al. [12] combined email metadata with email content using BM25F. Ogilvie and Callan [28] 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. Weerkamp et al. [33] explored incorporating thread, mailing list, and community content levels for email ranking.

Efficient search and email management strategies help people be more productive as they interact with communications. In this work, we explore how we can efficiently identify intents in enterprise email communications. This can further improve information workers productivity and enables creating new intelligent experiences to assist users with their tasks seamlessly and efficiently.

2.2 Email Intent Understanding

Previous research studied email acts and email intent analysis [4, 9, 24, 32]. Cohen et al. [9] 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 [5] 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 on an email message. Dabbish et al. [14] examined people's ratings of message importance and the actions they took on specific email messages with a survey of 121 people. Recently, Lin et al. [26] proposed using a reparametrized recurrent neural network to model actions that the recipient of the email might take upon receiving it. Lampert et al. [24] studied the problem of identifying messages that contain requests. They show that they can achieve better performance by segmenting the content of email messages into different functional zones (e.g. greetings, quoted text , etc.) and then considering only content in a small number of message zones. Bennett and Carbonell [4] studied the problem of action item detection from email messages. They argue that unlike standard topic-driven text classification, action-item detection requires inferring the intent of the sender, and identifying the sentence that directly indicates the action item.

Our work differs from the previous work in this area in several important ways. We extend the work of [14] by studying the different uses of email using large scale analysis of a publicly available dataset of enterprise email communications. We extend the work of [24] and [4] by expanding the notion of intent beyond action items and studying how to leverage context to improve intent detection.

2.3 Email Classification and Mining

Beyond email intent understanding, prior work has studied several other classification and mining tasks over email. Klimt and Yang [22] introduced the Enron corpus as a dataset and used it to explore automated classification of email messages into folders. Bekkerman et al. [3] extended this work by discussing the challenges that arise from differences between email foldering and traditional document classification. Pal and McCallum [30] proposed a model for suggesting who to add as an additional recipient for an email under composition. Graus et al. [18] generalized this to the task of recipient recommendation by leveraging both email content and communication graph signals. Another line of work focused on predicting reply behavior in email. On et al. [29] studied the problem of email reply order prediction by mining interaction behaviors. Kooti et al. [23] 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. [35] presented a detailed study for reply behavior in enterprise email and proposed methods for predicting whether a message will receive a reply and when the reply will occur. DiCastro et al. [15] studied four common user actions on email (read, reply, delete, delete-without-read) using a sample of 100k users of the Yahoo! Mail service. They proposed and evaluated a machine learning framework for predicting these four actions. We focus on extracting intents whose automated identification can be used to assist the user in performing intent-related tasks.

In the next section, we study the distribution of different intents in enterprise email communications and the advantage of contextual information in detecting those intents.

3 INTENTS IN ENTERPRISE EMAIL

We start by characterizing email intents in enterprise email and studying the effect of using context on detecting those intents. First, we define several categories of email intents that have been studied in previous work and study how they are manifested in enterprise email communications by analyzing a sample of a large scale enterprise email dataset. Second, we seek to understand whether using context could impact human understanding of email intent. To this end, we conduct an analysis to study whether humans benefit from context in understanding intent in email. We presume that to understand the meaning of text, humans not only read the given text but also pay attention to relevant information in the text surrounding it. Thus, before we study how to build models to leverage context, we focus on quantifying the impact of context on human ability to identify email intent. We start by adopting a categorization of different intents in email conversations based on previous work and provide an empirical analysis of intents in emails using a publicly available email collection.

3.1 Characterizing Types of Intents in Email

The objective of this analysis is to characterize the different types of intents that occur in email, how often they occur and how often they co-occur in a single message.

Following the work in [8, 13, 32], we define four distinct non-comprehensive categories for email intents including information exchange, task management, scheduling and planning, and social communication. Each of these categories can be associated with several intents. We provide definitions for each category and its sub-intents below:

Information Exchange: Information exchange intent involves communicating about information; either the sender intends to share information or to seek information. A common use of email includes asking questions, requesting or sharing content, status updates, etc. We define two sub-intents of this category: *share information* and *request information*. Sharing information means the sender would like to share information or content with the recipient(s), such as FYI messages, progress updates, status updates, or documents. Requesting information denotes a scenario where the sender is requesting information that can be potentially responded to by sharing a document or a similar resource. Note that these sub-intents are not comprehensive. For example, there are other intents that can be associated with seeking information such as asking questions, requesting confirmation, etc.

Task Management: Email is often used to manage tasks and the actions associated with those tasks. The definition of task management is very general and thus can be divided into two distinct sub-intents: *request action* and *promise action*. Requesting an action means that the sender is asking the recipient to perform some activities and promising an action means the sender is committing to perform an action.

Scheduling and Planning: Scheduling and planning over email involves the scenario where people intend to schedule an event or share a reminder about a coming event. This category of intents includes *schedule meeting* and *reminder*. Scheduling meeting refers to the sender's intention to organize an event, such as a physical meeting, a phone call or a conference call. Reminder refers to the sender's intention to remind the recipient about an upcoming event.

Social Communication: Social communication are casual messages such as greeting messages or thank you notes, which are exchanged between friends and family, as well as work contacts. Examples of sub-intents for the social communication category include, but are not limited to, *greeting messages* and *thank you notes*.

To better understand the characteristics of user intent in email messages, we launched an annotation task to manually annotate the intents in the Avocado[1] research email collection [27] from the Linguistic Data Consortium. This collection contains corporate emails from a defunct information technology company referred to as "Avocado". The collection contains an anonymized version of the full content of emails, and various meta information from Outlook mailboxes for 279 company employees. The full collection contains 938, 035 emails. We selected a total of 1300 email threads uniformly at random from the Avocado dataset and annotated the first email message by three annotators according to the intents and sub-intents discussed earlier in this section.

For each message, the annotator could choose multiple intents and the final judgment is made by a majority-voting strategy. In our

[1]Some in the research community view Avocado as a more appropriate research test bed than the Enron collection since Avocado entered the public domain via the cooperation and consent of the legal owner of the corpus while Enron entered via a legal discovery process with no particular consent process for use in research.

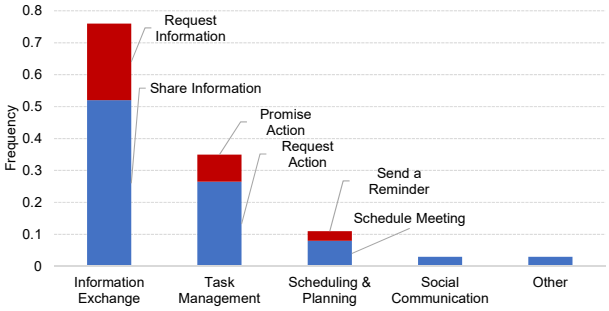


Figure 2: Frequency of each intent in a subset of the Avocado dataset (broken by sub-intent as described in Section 3.1)

case, the final intents for the email are those which are selected by at least two annotators. We achieve a substantial Kappa score 0.694 for the inter-annotator agreement. Figure 2 shows the distribution of the emails over sub-intents in the Avocado dataset.

Note that our list of sub-intents is not comprehensive. Hence, we allowed an annotator to select *Other* when they were unable to identify the intent (e.g. a forwarded message with no text) or when the emails had an intent that is not defined in our list.

Distribution of Intent Types: Figure 2 shows different intents and the frequency by which they appear in our sample of annotated messages. The figure shows that *information exchange* and *task management* are the most frequent intents in enterprise emails. This is similar to findings in previous work [13] that used surveys to characterize different usage of email. We also note that the percentage of the *information exchange* intent is significantly higher in our analysis while the *scheduling* intents are considerably lower compared to the survey results in [13]. The discrepancy can be explained by the differences in the method of data collection, data source, and the amount of annotated data. Note that our analysis is based on annotation of a public email data set where email messages where selected at random. While in [13], the data was collected for 581 emails through surveys which asked respondents to provide intent information about five new non-spam messages in their email inbox. It is possible that when we ask respondents to recall an email, they tend to think about emails that may not match the distribution of all emails in their mailboxes. Additionally, different work environments (a university in the case of [13] and an IT company in our case) could affect the distribution of intents in email messages.

Single vs. Multiple Intents: Emails usually contain more than one intent and the intents are not mutually exclusive. For example, an email message could be sending a reminder about a deadline and requesting an action to be completed before the deadline. Our analysis shows that approximately 55.2% of messages contain a single intent, 35.8% contain two intents and 9.0% contain three or more intents. Looking in more details at the emails with multiple sub-intents, we observed that some intents are highly correlated. Figure 3 shows the co-occurrence of different sub-intents in the same email which is based on the frequency of emails with specific pairs of sub-intents. In addition, we can observe that *share information* and *request information* are very likely to happen in the same email, while *social* and *reminder* or *schedule meeting* sub-intents are unlikely to co-exist in the same email. An example of an email

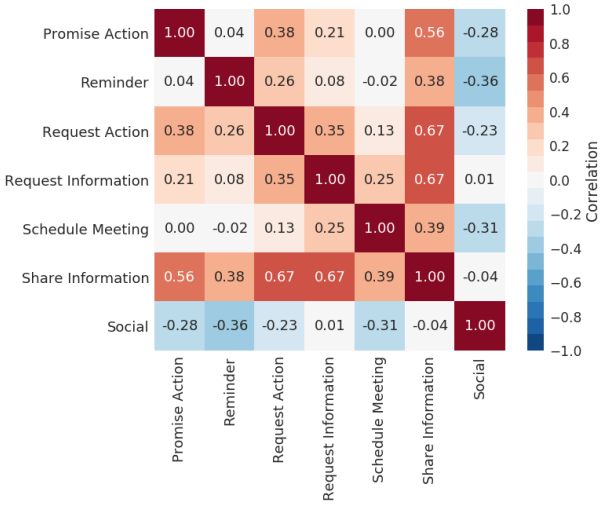


Figure 3: The distribution of sub-intent pairs in the same email

message which contains *share information* and *request information* sub-intents is *"We have been invited to the X conference in Seattle. Please let me know if the team has enough budget to pay for our travel"*.

This section provided a brief analysis of intent distribution in an enterprise email collection. This analysis was intended to serve two main purposes. First, it provided the necessary information to guide the selection of a number of intents to focus on for the intent identification study (see Section 5.1). Second, it verified that following previous work by identifying intents at the sentence-level (versus assigning a single intent to the whole message) is a reasonable choice given the fact that close to half of the messages contained more than one intent.

Now we turn our attention to how these intents can be detected from email text. But before we discuss how machine learning models could be used to accomplish this task, we first seek to test our hypothesis that context helps with identifying intent by conducting a study where we compare the performance of humans in identifying intent when they have access or lack access to context. The study and the results are described in the next subsection.

3.2 Effect of Contextual Information on Human Performance

The main focus of this work is to study the effect of leveraging contextual information for email intent detection. Before jumping to building machine learning models for intent identification, we study how helpful contextual information can be for a human annotator. We use one intent as an example, the *request information* intent, and study the effect of contextual information on human performance in identifying whether a sentence contains the intent or not. As explained early, we use the the whole email body as context and seek to understand whether it will help human annotators to make better judgments on a target sentence. We picked 540 instances from the ground truth set such that half of them have positive labels (containing the *request information* intent). We sent all instances to two groups of human annotators (crowdworkers). The annotators

in these two groups do not have any overlap. One of the group had access to the full email body, with the target sentence highlighted, and the other only had access to the target sentence. Each instance is annotated by three people and the majority of the annotations for each sentence represents its human prediction. We can use the annotations as predictors and calculate the human predictors precision and recall in the two settings by comparing annotations with ground truth labels.

Table 1: Positive Precision and Recall for human prediction where context is available (With Email Body) and context not available (Without Email Body)

Annotation Task	Precision	Recall
With Email Body	93.45	76.61
Without Email Body	91.51	52.86

Table 1 shows drops in both positive precision and recall when the human annotators were not provided with the email body. However, the cutback in positive precision is not as significant as the cutback in positive recall. To make sure that the comparison between these two annotations is fair and not biased toward the annotation behavior, we calculated the Krippendorff's α agreement score [20] for inter-annotator agreement among multiple judges. The α score for the annotation task including the full body of the email message is 0.58 and for the task without the email body is 0.56. Note that the α scores are only slightly different which implies the annotation behavior does not vary substantially from one task to the other.

To further understand the impact of context on human predictors, we show the confusion matrix for human predictions without context in Table 2. This provides more information about the type of errors humans do when they lack access to the context. Recall that we defined human predictions for each sentence as the majority of the annotations for each sentence when the annotators do not have access to the email body.

Table 2: Confusion Matrix for Human Predictions

Predictions	True Positive	True Negative
Positive	175 (%32.4)	14 (%2.6)
Negative	95 (%17.6)	256 (%47.4)

Table 2 shows that the true positive sentences benefit significantly from the contextual information in the email body. Overall, this analysis shows that human annotators perform far better in intent detection when they are provided with contextual information, in the form of the full body of the email message. Similarly, we can argue that machine learning models for identifying intent can also benefit from the contextual information. In next section, we will discuss how we incorporate the context information into a linear model (i.e. logistic regression) and deep learning model (i.e. a recurrent neural network) to improve sentence level intent detection in email.

4 CONTEXT-AWARE USER INTENT IDENTIFICATION

The main task we pursue in this paper can be defined as follows: given an email message and a sentence in that email, we aim to predict whether the sentence implies a given intent or not. We have shown earlier that humans benefit from the context provided by the full email body when identifying intents. As such, we propose a simple framework, shown in Figure 4, to leverage the context information for sentence-level intent identification. The framework consists of three major components: sentence encoder, context encoder and feature fusion layer. The sentence encoder takes as input the target sentence and aims to extract the local features from the sentence itself. Similarly, the context encoder would extract the global features from the full body. The outputs of the sentence and context encoders are then fed into the feature fusion layer to generate the context-aware sentence representation, which is used for the final intent classification. The feature fusion layer could be considered as an aggregation function of the local sentence features and the global context features. The framework allows us to extract features from the target sentence, the context (which was shown to be useful in the analysis in Section 3.2) and augmenting the two feature sets together. The framework could be used in a traditional machine learning model or a deep learning model. We describe each in turn next.

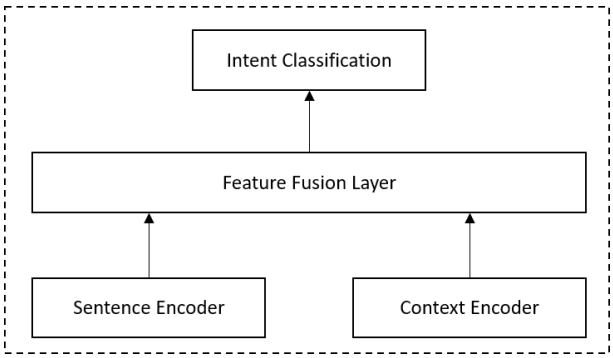


Figure 4: Context Augmented Sentence-Level intent identification framework

4.1 A Traditional Machine Learning Model

Previous works [4, 11] have used learning algorithms such as logistic regression and Support Vector machines for detecting intents in email text. We can extend this work to incorporate context following the framework in Figure 4 as follows:

Sentence Encoder: The sentence encoder would be designed as a handcrafted feature extractor. We generate n-grams (up to 3-grams) from the target sentence and use n-gram TF-IDF values as features.

Context Encoder: Just like the sentence encoder, the context encoder is designed as a handcrafted feature extractor with n-grams (up to 3-grams) extracted from the full message body.

Feature Fusion: To augment the sentence features with context features, we concatenate the sentence and context feature vectors. This allows the classifier access to features extracted from the context while maintaining separate feature spaces for the target sentence and the context.

Intent Classification: The concatenated feature vector is fed to a classifier. We experiment with both Logistic Regression (LR) and Support Vector Machines (SVM) for the classifier. Both learning

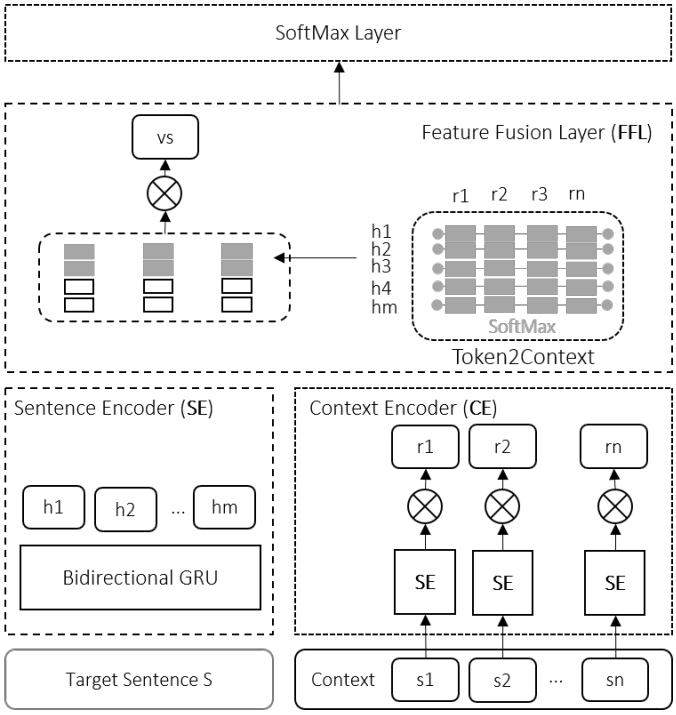


Figure 5: Overview of the DCRNN model structure: Sentence Encoder (SE) is a bi-directional GRU, which output the hidden states of the tokens in the sentence. The Context Encoder (CE) use the same SE for each sentence in the Context. $\otimes$ is the attention operator which takes the hidden states matrix as input and output a vector.

algorithms yield compact, interpretable models that have been previously used for task modeling on email (e.g., [10]).

4.2 A Deep Learning Model

Motivated by the recent advances in applying neural network methods to natural language understanding tasks, we use a neural network approach to better represent the text of the target sentence, the context and the interaction between them.

Sentence Encoder: Given a sentence s with a list of words w_i, $i \in 1..L$, the sentence encoder aims to enrich the representation of each word with knowledge from the sentence scope. For each word w_i in the sentence, we first transform them into dense vectors through a word embedding matrix [25] $W \in R^{d \times |V|}$. Here $|V|$ is the size of vocabulary, and d is the dimension of the word embedding. Then we apply a bi-directional recurrent neural network (RNN) with GRU cells [6] to the sentence s. The bi-directional RNN contains two RNNs, forward RNN and backward RNN, one reads the sentence with a forward order and the other reads it in reverse. We obtain the hidden state [6] h_i for each word w_i in sentence s by concatenating the forward hidden state $\overrightarrow{h_i}$ and the backward hidden state $\overleftarrow{h_i}$, i.e., $h_i = [\overrightarrow{h_i}, \overleftarrow{h_i}]$. Scanning the text from both directions allows the representation of each word to carry information from the words before and after it.

Context Encoder: Given a context (i.e. full message body) c with a list of sentence s_j, $j \in 1..N$, the context encoder aims to encode each sentence in the context to a fixed vector space. We first apply the same sentence encoder as we used to encode each sentence in the context. Given the hidden states of the words in the sentences, there are several ways to build the sentence representation, such as max-pooling and averaging over the hidden state matrix. One disadvantage of these methods is that they treat all words of the sentence equally. However, we noticed that some keywords in the sentences tend to be more important than others for identifying the user intent. For example, *meet, discuss, get together* are strong signals for the schedule meeting intent. As such, we chose to apply an attention operation [2] on the sentence and use the weighted average of the hidden states as the sentence representation. The more important words are expected to have larger attention weights which can be learned from the data during model training. To compute the attention weights, we define a sentence status vector u_s where the dimension of u_s is a hyperparameter we can set arbitrarily. The attention operation takes all the hidden states H as input, and outputs the weight vector α as:

$$\alpha = Softmax(u_s Tanh(W_s H^T + b_s)) \tag{1}$$

Note that $H = [h_1, h_2, ..., h_L]$, and W_s and b_s are the weight matrix and bias vector of a one-layer MLP. The sentence representation r_s is achieved as:

$$r_s = \alpha H \tag{2}$$

Feature Fusion: Given the hidden states H_s of the target sentence s and the sentence representations R_c of the context c, we aim to augment every token in s with the relevant information from c. To accomplish this, we compute an unnormalized attention matrix $A = F(H_s, R_c) \in R^{L \times N}$, here L is the length of the target sentence s and N is the number of sentences in the context c. Each element in A_{ij} is a scalar that is intended to represent the relation between the token i in the target sentence and the sentence j in the context. Each element A_{ij} is computed as

$$A_{ij} = w_c^T [H_s^i; R_c^j; H_s^i \circ R_c^j] \in R \tag{3}$$

where w_c is trainable weight vector, $[;]$ is vector concatenation and $\circ$ is element-wise multiplication. This results in a matrix A where every row represent a token in the target sentence and each element represents how relevant each sentence in the context is to this token. We then normalize the values of each row of A (such that they add up to 1) to generate a vector a_l for each token in the target sentence. The context-aware representation for the l_{th} token is represented as $H_s^l = [H_s^l; (R_c)^T a_l]$. Finally, we apply another attention operation similar to equations 1 and 2 on H_s to get the final context-aware representation v_s for the target sentence.

Intent Classification: Given the target sentence representation v_s, we then feed it a single softmax layer function to perform the prediction. It yields:

$$p = Softmax(W_p v_s + b_p) \tag{4}$$

where p is the prediction probability, and W_p, b_p are the parameters of the final full connection layer. We use cross-entropy loss to train the model.

The full model is illustrated in Figure 5 and is referred to as a Dynamic-Context Recurrent Neural Network (DCRNN) model.

Table 3: Examples of the thee intents types used for experiments: *Request Information, Schedule Meeting* and *Promise Action*

Intent Type	Examples
Request Information	Can anyone point me to the specs document of project Blue?
Schedule Meeting	It would be great to get together to discuss the project status when you are back.
Promise Action	I will create the slides for the project review next week and share them for feedback.

5 EXPERIMENTS

5.1 Tasks

Following previous work and the analysis presented in Section 3, we experiment with the following three intents:

(1) **Request Information**: The sender is requesting information that can be potentially responded to by sharing a document or a similar source;
(2) **Schedule Meeting**: The sender is expressing the desire to meet or suggesting a meeting that can potentially be scheduled and added to the participants calendars ;
(3) **Promise Action**: The sender is promising to perform an action that can potentially be added to her to-do list.

Note that the selected intents represent all the categories discussed in Section 3 except the social communication categories. Also note that the intents can all be directly linked to actions than can be taken on the message. For example, a *request information* intent can result in a document being shared, a *schedule meeting* intent can result in a calendar item being created on the participants calendars, and a *promise action* intent can result in an item added to the sender's to-do list. The proposed models should be easily extended to new intents once training data for more intents are available.

The intents were annotated by human annotators who examined the entire email and determined whether it has a given intent or not. Additionally, the text span, within email, where the intent is manifested was highlighted. As discussed in the analysis in Section 3, multiple intents could coincide in one message. Note that the negative examples were selected in a way that makes the prediction problem more challenging. For each intent, the negative examples were limited to the sentences that were not labeled as positive but contained one or more words from the list of top 100 words (according to TF-IDF) which frequently appeared in the positive sentences. For example, for the *schedule meeting* intent, the list contained words like "meet", "schedule", "discuss", etc. Additionally, text that appears as quoted text from previous messages, greetings and signatures was excluded. Each instance was labeled by 3 annotators and a majority vote was used to determine the final label. The Cohen's kappa value for all tasks was larger than 0.6 showing a substantial agreement between annotators. The sizes of the datasets were 47914, 9076, and 7080 for the *schedule meeting*, *promise action* and *request information* respectively. The positive instances ratio was 15.5%, 28.5% 19.6% respectively. Table 3 shows examples of different types of intents.

5.2 Evaluation

To evaluate the model, we split the full dataset of each task to training, validation, and test datasets using 5-fold cross validation. We split the dataset such that 60% of the data is used for training, 20% for validation and 20% for testing. Data was split based on user identifiers such that data from any given user would belong to either training, validation, or test data. We used the validation set to tune all hyper-parameters (e.g., L1 and L2 weights for LR, batch size, learning rate, dropout rate, and GRU hidden unit size for deep learning). We use the test data for evaluation and use F1 score as our main metric. For the deep learning methods, We use the pre-trained 300 dimension Glove vectors as the initial word embedding. We tune the hyper-parameters: batch size (32, 64), learning rate (0.1, 0.001, 0.0001), GRU hidden unit size(10, 30, 50) and dropout rate (0.2, 0.5, 0.7) based on the validation set. For training, we use the Adam Optimization Algorithm and initialize the model parameters using the method in [17].

5.3 Results

Overall Results: To evaluate the effect of adding context, we use several sentence-only models as baselines to several models that try to leverage both the target sentence and the context. We use F1-Score as evaluation metric and the Statistical significance is tested using a paired t-test with $p<0.05$ indicating significance. For the sentence only models, we follow previous work [4, 11] by training logistic regression (LR) and Support Vector Machine (SVM) models on n-gram features of the text. We also add strong text classification baselines that use Convolutional Neural Networks (CNN) for sentence classification [21]. For the models that use both the sentence and the context, we use the two traditional machine learning models described in Section 4.1 (LR: Sentence + Context and SVM:Sentence + Context) and the deep learning model described in Section 4.2 (DCRNN). The results of all methods on the three tasks described earlier are shown in Table 4. The table shows that given the sentence as the only input, the SVM and the CNN model tend to perform better with the first achieving the best results on the *Schedule Meeting* and *Request Action* tasks while the latter achieves the best results on the *Promise Action* task.

Table 4: Classification results F1 comparing sentence only models and sentence + context models for different intent-identification tasks: Schedule Meeting (SM), Promise Action (PA) and Request Information (RI). ‡ indicates statistically significant improvement over all compared approaches.

	SM	PA	RI
LR : Sent	66.86	74.73	74.71
SVM : Sent	66.24	73.56	75.45
CNN : Sent	67.12	75.21	73.15
LR : Sent + Cont	66.302	74.75	74.76
SVM : Sent + Cont	64.24	71.53	75.20
DCRNN	**73.48‡**	**80.42‡**	**78.37‡**

We hypothesized that the context information along with the target sentence would improve the performance of intent identification

Table 5: Classification results F1 for different intent-identification tasks for variants of the DCRNN mode: *DCRNN: Concat* uses a simpler fusion layer where the output of the sentence and context decoders is concatenated, *Sentence Decoder Only* ignore the context while *Context Encoder Only* uses only the context encode and ignores the target sentence. ‡ indicates statistically significant improvement over all compared approaches.

	SM	PA	RI
DCRNN	73.48‡	80.42‡	78.37‡
DCRNN: Concat	71.244	78.23	76.25
Sent Encoder Only	69.21	76.19	74.11
Cont Encoder Only	43.29	61.33	67.12

at the sentence level. It is interesting to see that the performance of the LR and SVM models do not show consistent improvement over the sentence-only model across tasks. More specifically the performance of LR model (Sentence + Context) is very close to the performance of the sentence only LR model. The SVM model shows significant drop for the *schedule meeting* task (2 points drop) and the *promise action* task (2 points drop). One possible explanation for the performance drop of LR and SVM models is that the LR and SVM model failed to pay attention to the relevant information in the context (i.e. full message body) while ignoring the irrelevant information. Another reason might be that the concatenation of the features from the target sentence and the context is not representative enough to capture the interaction between them (the context is represented with the same features regardless of the target sentence). More effort on feature engineering may be needed to enable the LR and SVM model to leverage the context information.

On the other hand, the DCRNN model does a much better job when leveraging the context information achieving significant gains for all three tasks. Unlike the LR and SVM model, the DCRNN model is able to model the interaction between the target sentence and the context. Moreover, it is able to pay more attention to the relevant parts of the context for each token in the target sentence.

Effect of different model components: As shown in Figure 4, the DCRNN consists of several components. To understand the impact of each component on the model performance, we remove different components, one at a time, and observe the impact on the performance compared to the full model. The results are shown in Table 5. We start by removing the fusion layer and replacing it with a simple concatenation of the output of the sentence encoder and the context encoder (*DCRNN:Concat)*. We notice a drop in performance on all tasks showing that going beyond concatenation has a positive impact on the overall performance. This shows that different parts of the context could be more important to consider for different tokens of the target sentence. Note that the feature fusion component, described in Section 4.2, ensures that the context representation is conditional on and relevant to the target sentence. On the other hand, concatenation results in the same context representation used regardless of the sentence. Next, we experiment with dropping the context encoder (*Sentence Encoder Only)* and the sentence encoder (*Context Encoder Only)*. Note that when we drop one of the components, we still apply an attention operation (see

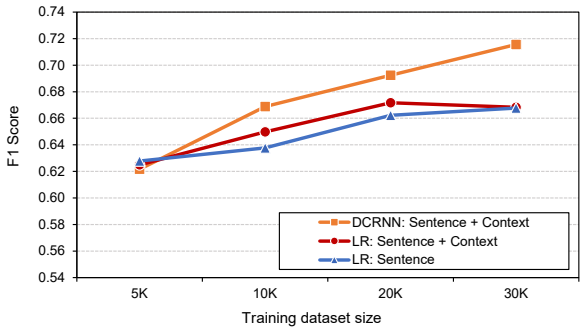


Figure 6: The performance (F1-Score) of models with different training data sizes (Schedule Meeting task)

equations 1 and 2) on the output of the remaining encoder and then feed the resulting vector to the classifier. We observe a significant drop in the performance in both cases. The drop is significantly bigger when only the context is used. This highlights both the value of the context and also that it only serves to augment the information in the target sentence and cannot replace it.

Effect of the training data size: We study the effect of training data size for non-deep learning and deep learning based approaches on *schedule meeting* task. We choose LR model (both using sentence only and sentence + context) and the DCRNN model for this study, we omit the SVM model since it had similar performance to the LR model. We fixed the size of validation and test set, and generated three new training sets by randomly sampling 5K, 10K and 20K instances from the original training set. Figure 6 shows the performance of different algorithms trained on different dataset sizes. In the case 5K training data, the performance of all models is almost identical. As the training data size increases, the performance of all models improves. Additionally, the the DCRNN models start to outperform the LR model with an increasing performance gap.

Effect of the context size: So far we have been using the word *context* to refer to the full email body where the target sentence is located. Another way to define *context* is the surrounding text of the target sentence with a specific window size. Figure 7 shows the performance of the DCRNN model for different context window sizes. We use *0* to refer to the case where only the target sentence is used as input to the model and $\pm$ *n* to refer to the case where the *n* sentences before and after the target sentence are used as its context. We observe that there is an increasing trend in F1-score along with the increasing size of the context windows for all three tasks. The models get the best performance when we use a window size of the full email body as the context. Benefiting from the attention mechanism, our approach could be able to handle the noisy information and capture the dependency between the target sentence and other sentences, even if these sentences are not adjacent to the target sentence.

Effect of User Interaction History and Temporal Metadata: There are other contextual information about an email message that we can leverage, including user interaction history and email

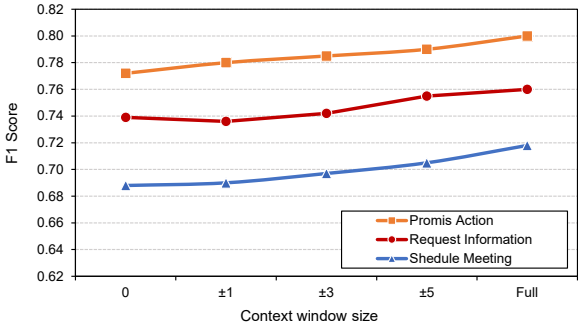


Figure 7: The performance(F1-Score) of DCRNN model with different context sizes

Target Sentence:
So let's make it 3pm.

No	Weight	Context Sentence
1	0.4025	My 3pm meeting canceled earlier today.
2	0.1647	So let's make it 3pm.
3	0.4328	I will call you on your office line.

Figure 8: An example showing the most relevant sentences from the full body corresponding to the target sentence. Relevance is computed by aggregating the attention weights for the context sentences over tokens in the target sentence.

metadata and temporal information. These features were extensively studied in the literature in context of tasks like email reply prediction. We follow the same definition of features used for reply prediction in the literature and use the same features defined in [35]. We picked the *request information* intent identification task and studied the effect of adding these features on the performance of a model that uses content only. We split the data into training, validation, and test sets using the arrival time of emails. This ensures we mimic the real world scenario where we only have access to historical data from the past and we test them on future data. We examined the performance of the LR model with different sets of contextual features and observed that the users' interaction history and emails metadata and temporal information have almost no effect on the performance of our model and therefore we did not leverage this information further in our analysis. Further work is needed to define other user history and metadata features that may be more appropriate for these tasks.

5.4 Case Study

To understand how the context information helps the intent identification on sentence level, we look into the attention weights on the tokens in the target sentence and the sentences in the context. We notice that our DCRNN model is able to pay attention to supporting information from the context which could be very useful when the target sentences are ambiguous. For instance, with only

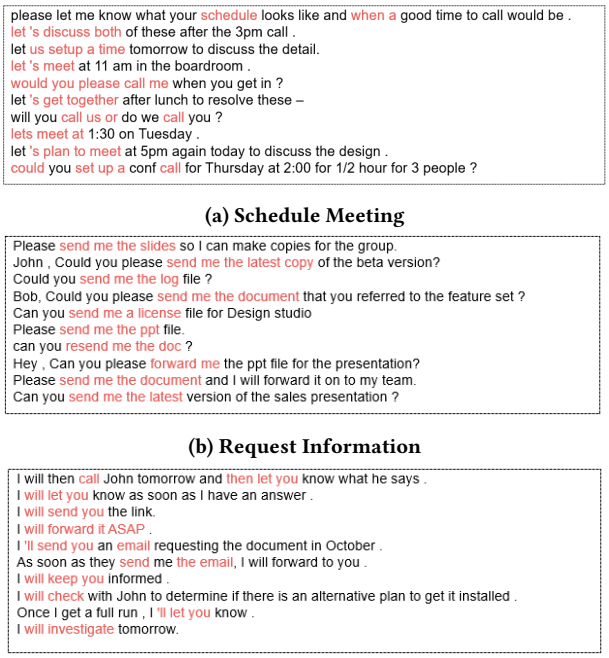


Figure 9: An example showing the most important words (according to the token attention weights for each token in the target sentence) for multiple sentences.

the target sentence "*So let's make it 3pm*" in Figure 8, it would be hard to decide the intent of the user. The user might intend to set up a meeting time with the recipient or just assign a deadline for a task. As shown in Figure 8, the model assigned high weight to the two keywords "*let's make*" (orange highlighted) in the target sentence and these two keywords put more attention on sentence 1 and 3 in the context. In sentence 1 and sentence 3, the model focused more on "*meeting canceled*" and "*will call you*". Along with the selected context information, it is more clear that the target sentence "*So let's make it 3pm*" in this case pertains to the *schedule meeting* intent.

To further analyze the patterns learned by the model, in Figure 9, we visualize the attention weights of the tokens from 10 sentences with the positive prediction on the high confidence level for each task. We highlight the tokens with attention weights higher than 0.1 in each sentence. We find our model assigns higher weights to intent-related keywords. For instance, in the *schedule meeting* model, the meeting related keywords such as *schedule*, *discuss* and *get together* are being highlighted. In the case of *request information*, critical words like *send*, *forward*, *document* and *slides* receive more attentions. And in the *promise action* task, the typical patterns like *"will + verb"*, *"let you"* and *"keep you"* are captured by the model.

6 CONCLUSIONS

In this paper, we studied the problem of user intent understanding in workplace email. We studied a large scale publicly available email dataset to characterize intents in enterprise email. We showed

that a variety of intents occur in enterprise communication. Automatically detecting these intents could not only help us to better understand communications in the workplace, but also allow us to create new experiences that assist the users with completing tasks and retrieving information efficiently. Typical approaches to email intent understanding have focused on assigning broad categories to the whole message or on classifying sentences one at a time. We showed that sentence-level classification tends to ignore additional context provided in the email message. To study this further, we conducted a study where we asked human annotators to annotate sentences in context and in isolation. We showed that the ceiling of performance is much lower for humans if context is discarded. This inspired us to study how to incorporate context in automatic classification of intent in email. We showed that incorporating context within a model using n-gram features with methods such as logistic regression or support vector machines fails to show significant improvements over sentence-level models. Hence, we proposed a neural network based approach using a context-aware attention mechanism. We showed that the proposed model can significantly improve the performance by leveraging the context. Comparing these gains to the human results indicates that we have not yet reached the maximum benefit that can be realized by leveraging context. Our future work will aim to develop better models to realize the full potential of leveraging context and will extend the notion of context to cover thread-level information, users' history, and metadata about email messages.

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. *CoRR* abs/1409.0473 (2014). arXiv:1409.0473 http://arxiv.org/abs/1409.0473

[3] Ron Bekkerman, Andrew Mccallum, and Gary Huang. 2004. Automatic categorization of email into folders: Benchmark experiments on Enron and SRI corpora. (01 2004).

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

[5] 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.

[6] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[7] 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).

[8] William Cohen, Vitor Carvalho, and Tom Mitchell. 2004. Learning to Classify Email into "Speech Acts". In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*.

[9] 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*.

[10] Simon Corston-Oliver, Eric Ringger, Michael Gamon, and Richard Campbell. 2004. Task-focused summarization of email. *Text Summarization Branches Out* (2004).

[11] Simon Corston-Oliver, Eric Ringger, Michael Gamon, and Richard Campbell. 2004. Task-focused summarization of email. In *ACL*.

[12] 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

[13] 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

[14] 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.

[15] 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.

[16] 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.

[17] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. 249–256.

[18] David Graus, David van Dijk, Manos Tsagkias, Wouter Weerkamp, and Maarten de Rijke. 2014. Recipient Recommendation in Enterprises Using Communication Graphs and Email Content. In *SIGIR '14*. ACM, New York, NY, USA, 1079–1082.

[19] 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

[20] A. F. Hayes and K Krippendorff. 2007. Answering the call for a standard reliability measure for coding data. In *Communication Methods and Measures 1*. 77–89.

[21] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014).

[22] Bryan Klimt and Yiming Yang. 2004. The Enron Corpus: A New Dataset for Email Classification Research. In *ECML'04*. 217–226.

[23] 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.

[24] 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.

[25] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *ICML*. 1188–1196.

[26] 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).

[27] Douglas Oard, William Webber, David Kirsch, and Sergey Golitsynskiy. 2015. Avocado Research Email Collection. DVD. (2015).

[28] P. Ogilvie and J. Callan. 2005. Experiments with language models for known-item finding of e-mail messages. In *TREC*.

[29] Byung-Won On, Ee-Peng Lim, Jing Jiang, Amruta Purandare, and Loo-Nin Teow. 2010. Mining Interaction Behaviors for Email Reply Order Prediction. In *Proceedings of the 2010 International Conference on Advances in Social Networks Analysis and Mining (ASONAM '10)*. IEEE Computer Society, Washington, DC, USA, 306–310.

[30] Christopher Pal and Andrew McCallum. 2006. CC Prediction with Graphical Models. In *CEAS*.

[31] 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

[32] Maya Sappelli, Gabriella Pasi, Suzan Verberne, Maaike de Boer, and Wessel Kraaij. 2016. Assessing e-mail intent and tasks in e-mail messages. *Information Sciences* 358 (2016), 1–17.

[33] Wouter Weerkamp, Krisztian Balog, and Maarten Rijke. 2009. Using Contextual Information to Improve Search in Email Archives. In *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval (ECIR '09)*. Springer-Verlag, Berlin, Heidelberg, 400–411. https://doi.org/10.1007/978-3-642-00958-7_36

[34] 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

[35] 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.