Smart To-Do: Automatic Generation of To-Do Items from Emails

Sudipto Mukherjee[†*] **Subhabrata Mukherjee**[‡] **Marcello Hasegawa**[‡]
Ahmed Hassan Awadallah[‡] **Ryen White**[‡]
[†]University of Washington, Seattle [‡]Microsoft Research AI
sudipm@uw.edu,
{submukhe, marcellh, hassanam, ryenw}@microsoft.com

Abstract

Intelligent features in email service applications aim to increase productivity by helping people organize their folders, compose their emails and respond to pending tasks. In this work, we explore a new application, Smart-To-Do, that helps users with task management over emails. We introduce a new task and dataset for automatically generating To-Do items from emails where the sender has promised to perform an action. We design a two-stage process leveraging recent advances in neural text generation and sequence-to-sequence learning, obtaining BLEU and ROUGE scores of 0.23 and 0.63 for this task. To the best of our knowledge, this is the first work to address the problem of composing To-Do items from emails.

1 Introduction

Email is one of the most used forms of communication especially in enterprise and work settings (Radicati and Levenstein, 2015). With the growing number of users in email platforms, service providers are constantly seeking to improve user experience for a myriad of applications such as online retail, instant messaging and event management (Feddern-Bekcan, 2008). Smart Reply (Kannan et al., 2016) and Smart Compose (Chen et al., 2019) are two recent features that provide contextual assistance to users aiming to reduce typing efforts. Another line of work in this direction is for automated task management and scheduling. For example. the recent Nudge feature[1] in Gmail and Insights in Outlook[2] are designed to remind users to follow-up on an email or pay attention to pending tasks.

Smart To-Do takes a step further in task assistance and seeks to boost user productivity by automatically generating To-Do items from their email

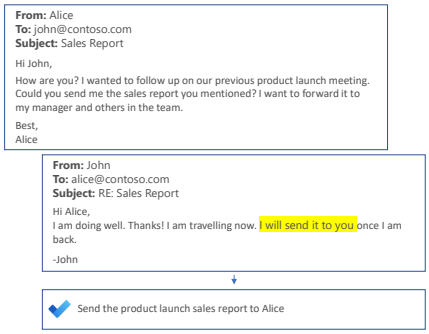

Figure 1: An illustration showing the email and a commitment sentence (in yellow) and the target To-Do item, along with other email meta-data.

context. Text generation from emails, like creating To-Do items, is replete with complexities due to the diversity of conversations in email threads, heterogeneous structure of emails and various meta-deta involved. As opposed to prior works in text generation like news headlines, email subject lines and email conversation summarization, To-Do items are *action-focused*, requiring the identification of a specific task to be performed.

In this work, we introduce the task of automatically generating To-Do items from email context and meta-data to assist users with following up on their promised actions (also referred to as commitments in this work). Refer to Figure 1 for an illustration. Given an email, its temporal context (i.e. thread), and associated meta-data like the name of the sender and recipient, we want to generate a short and succinct To-Do item for the task mentioned in the email.

This requires identifying the task sentence (also referred to as a *query*), relevant sentences in the email that provide contextual information about the query along with the entities (e.g., people) associated with the task. We utilize existing work to identify the task sentence via a commitment classifier that detects action intents in the emails. Thereafter

[*]Work done as an intern at Microsoft Research.
[1] Gmail Nudge [2] Outlook Insights

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 8680–8689
July 5 - 10, 2020.

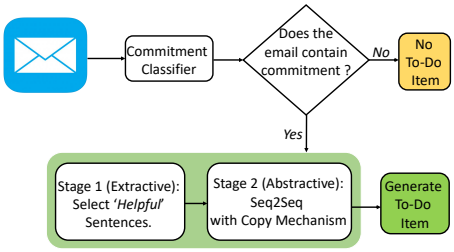

Figure 2: Smart To-Do flowchart: The email content is first scanned to detect any possible commitment sentence. If present, a To-Do item is generated using a two-stage Smart To-Do framework.

we use an unsupervised technique to extract key sentences in the email that are *helpful* in providing contextual information about the query. These pieces of information are further combined to generate the To-Do item using a sequence-to-sequence architecture with deep neural networks. Figure 2 shows a schematic diagram of the process. Since there is no existing work or dataset on this problem, our first step is to collect annotated data for this task. Overall, our contributions can be summarized as follows:

- We create a new dataset for To-Do item generation from emails containing action items based on the publicly available email corpus *Avocado* (Oard et al., 2015). [3]
- We develop a two-stage algorithm, based on unsupervised task-focused content selection and subsequent text generation combining contextual information and email meta-data.
- We conduct experiments on this new dataset and show that our model performs at par with human judgments on multiple performance metrics.

2 Related Works

Summarization of email threads has been the focus of multiple research works in the past (Rambow et al., 2004; Carenini et al., 2007; Dredze et al., 2008). There has also been considerable research on identifying speech acts or tasks in emails (Carvalho and Cohen, 2005; Lampert et al., 2010; Scerri et al., 2010) and how it can be robustly adapted across diverse email corpora (Azarbonyad et al., 2019). Recently, novel neural architectures have been explored for modeling action items in emails (Lin et al., 2018) and identifying intents in email conversations (Wang et al., 2019). However, there has been less focus on task-specific email summarization (Corston-Oliver et al., 2004). The closest to our work is that of email subject line generation (Zhang and Tetreault, 2019). But it focuses on a common email theme and uses a supervised approach for sentence selection, whereas our method relies on identifying the task-related context.

3 Dataset Preparation

We build upon the Avocado dataset (Oard et al., 2015)[4] containing an anonymized version of the Outlook mailbox for 279 employees with various meta-data and $938,035$ emails overall.

3.1 Identifying Action Items in Emails

Emails contain various user intents including planning and scheduling meetings, requests for information, exchange of information, casual conversations, etc. (Wang et al., 2019). For the purpose of this work, we first need to extract emails containing at least one sentence where the sender has promised to perform an action. It could be performing a task, providing some information, keeping others informed about a topic and so on. We use the term *commitment* to refer to such intent in an email and the term commitment sentence to refer to each sentence with that intent.

Commitment classifier: A commitment classifier $\mathcal{C} : \mathcal{S} \mapsto [0, 1]$ takes as input an email sentence $\mathcal{S}$ and returns a probability of whether the sentence is a commitment or not. The classifier is built using labels from an annotation task with 3 judges. The Cohen's kappa value is 0.694, depicting substantial agreement. The final label is obtained from the majority vote, generating a total of 9076 instances (with 2586 positive/commitment labels and 6490 negative labels). The classifier is an RNN-based model with word embeddings and self-attention geared for binary classification with the input being the entire email context (Wang et al., 2019). The classifier has a precision of 86% and recall of 84% on sentences in the Avocado corpus.

3.2 To-Do Item Annotation

Candidate emails: We extracted $500k$ raw sentences from Avocado emails and passed them

[3] We will release the code and data (in accordance with LDC and Avocado policy) at https://aka.ms/SmartToDo. Email examples in this paper are similar to those in our dataset but are not reproducing text from the Avocado dataset.

[4] Avocado is a more appropriate test bed than the Enron collection (Klimt and Yang, 2004) since it contains additional meta-data and it entered the public domain via the cooperation and consent of the legal owner of the corpus.

Ground-truth	Update our quarterly sales in the head-office financial database.
Annotation	Update our quarterly sales in the database.
Fluency	4 (Grammatically correct, follows structure of To-Do item.)
Completeness	1 (Which *database* ? Does not include additional details from email context.)
Ground-truth	Test the server for load fault on Friday morning PST and let Bob know the result.
Annotation	Testing on server load fault on Friday morning PST and let Bob know the result.
Fluency	2 (Grammatically incorrect; starts with 'ing' verb and deviates from structure.)
Completeness	4 (Explains the context and contains all keywords)

Table 1: Snapshot of qualitative analysis of human annotations for fluency and completeness.

through the commitment classifier. We threshold the commitment classifier confidence to 0.9 and obtained $29k$ potential candidates for To-Do items. Of these, a random subset of $12k$ instances were selected for annotation.

Annotation guideline: For each candidate email e_c and the previous email in the thread e_p (if present), we obtained meta-data like '*From*', '*Sent-To*', '*Subject*' and '*Body*'. The commitment sentence in e_c was highlighted and annotators were asked to write a To-Do item using all of the information in e_c and e_p.

We prepared a comprehensive guideline to help human annotators write To-Do Items containing the definition and structure of To-Do Items and commitment sentences, along with illustrative examples. Annotators were instructed to use words and phrases from the email context as closely as possible and introduce new vocabulary only when required. Each instance was annotated by 2 judges.

Analysis of human annotations: We obtained a total of 9349 email instances with To-Do items, each of which was annotated by two annotators. To-Do items have a median token length of 9 and a mean length of 9.71. For 60.42% of the candidate emails, both annotators agreed that the subject line was helpful in writing the To-Do Item.

To further analyze the annotation quality, we randomly sampled 100 annotated To-Do items and asked a judge to rate them on (a) *fluency* (grammatical and spelling correctness), and (b) *completeness* (capturing all the action items in the email) on a 4 point scale (1: Poor, 2: Fair, 3: Good, 4: Excellent). Overall, we obtained a mean rating of 3.1 and 2.9 respectively for fluency and completeness. Table 1 shows a snapshot of the analysis.

4 Smart To-Do : Two Stage Generation

In this section, we describe our two-stage approach to generate To-Do items. In the first stage, we select sentences that are *helpful* in writing the To-Do item. Emails contain generic sentences such as salutations, thanks and casual conversations not relevant to the commitment task. The objective of the first stage is to select sentences containing informative concepts necessary to write the To-Do.

4.1 Identifying Helpful Sentences for Commitment Task

In the absence of reliable labels to extract helpful sentences in a supervised fashion, we resort to an unsupervised matching-based approach. Let the commitment sentence in the email be denoted as $\mathcal{H}$, and the rest of the sentences from the current email e_c and previous email e_p be denoted as $\{s_1, s_2, \ldots s_d\}$. The unsupervised approach seeks to obtain a relevance score $\Omega(s_i)$ for each sentence. The top K sentences with the highest scores will be selected as the extractive summary for the commitment sentence (also referred to as the query).

Enriched query context: We first extract top τ maximum frequency tokens from all the sentences in the given email, the commitment and the subject (i.e., $\{s_1, s_2, \ldots s_d\} \cup \mathcal{H} \cup \text{Subject}$). Tokens are lemmatized and stop-words are removed. We set $\tau = 10$ in our experiments. An enriched context for the query $\mathcal{E}$ is formed by concatenating the commitment sentence $\mathcal{H}$, subject and top τ tokens.

Relevance score computation: Task-specific relevance score Ω for a sentence s_i is obtained by inner product in the embedding space with the enriched context. Let $h(\cdot)$ be the function denoting the embedding of a sentence with $\Omega(s_i) = h(s_i)^T h(\mathcal{E})$.

Our objective is to find helpful sentences for the commitment given by semantic similarity between concepts in the enriched context and a target sentence. In case of a short or less informative query, the subject and topic of the email provide useful information via the enriched context. We experiment with three different embedding functions.

(1) Term-frequency (Tf) – The binarized term

Algorithm	At-least One Helpful @ K=2	@ K=3
Tf	0.80	0.85
FastText (Mean)	0.76	0.90
FastText (Max)	**0.85**	**0.92**
BERT (Pre-trained)	0.76	0.89
BERT (Fine-tuned)	0.80	0.89

Table 2: Performance of unsupervised approaches in identifying helpful sentences for a given query.

frequency vector is used to represent the sentence.

(2) FastText Word Embeddings – We trained FastText embeddings (Bojanowski et al., 2017) of dimension 300 on all sentences in the Avocado corpus. The embedding function $h(s_j)$ is given by taking the max (or mean) across the word-embedding dimension of all tokens in the sentence s_j.

(3) Contextualized Word Embeddings – We utilize recent advances in contextualized representations from pre-trained language models like BERT (Devlin et al., 2019). We use the second last layer of pre-trained BERT for sentence embeddings.

We also fine-tuned BERT on the labeled dataset for commitment classifier. The dataset is first made balanced (2586 positive and 2586 negative instances). Uncased BERT is trained for 5 epochs for commitment classification, with the input being word-piece tokenized email sentences. This model is denoted as BERT (fine-tuned) in Table 2.

Evaluation of unsupervised approaches: Retrieving at-least one helpful sentence is crucial to obtain contextual information for the To-Do item. Therefore, we evaluate our approaches based on the proportion of emails where at-least one helpful sentence is present in the top K retrieved sentences.

We manually annotated 100 email instances and labeled every sentence as *helpful* or not based on (a) whether the sentence contains concepts appearing in the target To-Do item, and (b) whether the sentence helps to understand the task context. Inter-annotator agreement between 2 judgments for this task has a Cohen Kappa score of 0.69. This annotation task also demonstrates the importance of the previous email in a thread. Out of 100 annotated instances, 44 have a replied-to email of which 31 contains a *helpful* sentence in the replied-to email body (70.4%). Table 2 shows the performance of the various unsupervised extractive algorithms. FastText with max-pooling of embeddings performed the best and used in the subsequent generation stage.

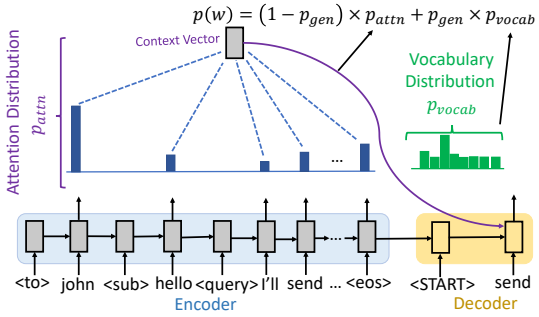

Figure 3: Seq2Seq with copy mechanism. Tokens involving named entities and task-specific keywords from the email are learned to copy in the To-Do item.

4.2 To-Do Item Generation

The generation phase of our approach can be formulated as sequence-to-sequence (Seq2Seq) learning with attention (Sutskever et al., 2014; Bahdanau et al., 2014). It consists of two neural networks, an encoder and a decoder. The input to the encoder consists of concatenated tokens from different meta-data fields of the email like 'sent-to', 'subject', commitment sentence $\mathcal{H}$ and extracted sentences $\mathcal{I}$ separated by special markers. For instance, the input to the encoder for the example in Figure 1 is given as:

<to> alice <sub> hello ? <query> i will send it to you < sent> could you send me the sales report ? <eos>

We experiment with multiple versions of the generation model as follows:

Vanilla Seq2Seq: Input tokens $\{x_1, x_2, \ldots x_T\}$ are passed through a word-embedding layer and a single layer LSTM to obtain encoded representations $h_t = f(x_t, h_{t-1}) \forall t$ for the input. The decoder is another LSTM that makes use of the encoder state h_t and prior decoder state s_{t-1} to generate the target words at every timestep t. We consider Seq2Seq with attention mechanism where the decoder LSTM uses attention distribution a_t over timesteps t to focus on important hidden states to generate the context vector h_t. This is the first baseline in our work.

$$\begin{aligned} e_{t,t'} &= v^T tanh(W_h \cdot h_t + W_s \cdot s_{t'} + b) \\ a_{t,t'} &= softmax(e_{t,t'}) \\ h_t &= \textstyle\sum_{t'} a_{t,t'} \cdot h_{t'} \end{aligned} \quad (1)$$

Seq2Seq with copy mechanism: As the second model, we consider Seq2Seq with copy mechanism (See et al., 2017) to copy tokens from important email fields. Copying is pivotal for To-Do item generation since every task involves named

From: John Carter *To:* Helena Watson; Daniel Craig; Rupert Grint *Subject:* Thanks Thank you for helping me prepare the paper draft for ACL conference. Attached is the TeX file. Please feel free to make any changes to the revised version. I sent to my other collaborators already and am waiting for their suggestions. I'll keep you posted. Thanks, John.
GOLD: Keep Helena posted about paper draft for ACL conference.
PRED: Keep Helana posted about ACL conference.
From: Raymond Jiang *To:*support@company.com *Subject:* Bug 62 Hi, there is a periodic bug 62 appearing in my cellphone browser, whenever I choose to open the request. It might be a JavaScript issue on our side, but it would be nice if you take a look. Thanks, Ray.
From: Criag Johnson *To:* Raymond Jiang *Subject:* Bug 62 Good Morning Ray, I shall take a look at it and get back to you.
GOLD: Take a look at Bug 62 and get back to Raymond.
PRED: Take a look at periodic and get back to Raymond.

Table 3: Generation example (GOLD: manual annotation, PRED: machine-generated) with email context.

Algorithm	BLEU-4	Rouge-1	Rouge-2	Rouge-L
Concatenate	0.13	0.52	0.28	0.50
Seq2Seq (vanilla)	0.14	0.53	0.31	0.56
Seq2Seq (copy)	**0.23**	**0.60**	**0.41**	**0.63**
Seq2Seq (BiFocal)	0.18	0.56	0.34	0.58
Human Judgment	0.21	0.60	0.37	0.60

Table 4: Comparison of various models for To-Do generation with BLEU and ROUGE (higher is better).

entities in terms of the persons involved, specific times and dates when the task has to be accomplished and other task-specific details present in the email context. To understand the copy mechanism, consider the decoder input at each decoding step as y_t and the context vector as h_t. The decoder at each timestep t has the choice of generating the output word from the vocabulary $\mathcal{V}$ with probability $p_{\text{gen}} = \phi(h_t, s_t, y_t)$, or with probability $1 - p_{\text{gen}}$ it can copy the word from the input context. To allow that, the vocabulary is extended as $\mathcal{V}' = \mathcal{V} \cup \{x_1, x_2, \ldots x_T\}$. The model is trained end-to-end to maximize the log-likelihood of target words (To-Do items) given the email context.

Seq2Seq BiFocal: As a third model, we experimented with query-focused attention having two encoders – one containing only tokens of the query and the other containing rest of the input context. We use a bifocal copy mechanism that can copy tokens from either of the encoders. We refer the reader to the Appendix for more details about training and hyper-parameters used in our models.

5 Experimental Results

We trained the above neural networks for To-Do item generation on our annotated dataset. Of the 9349 email instances with To-Do items, we used 7349 for training and 1000 each for validation and testing. For each instance, we chose the annotation with fewer tokens as ground-truth reference.

The median token length of the encoder input is 43 (including the helpful sentence). Table 4 shows the performance comparison of various models. We report BLEU-4 (Papineni et al., 2002) and the F1-scores for Rouge-1, Rouge-2 and Rouge-L (Lin, 2004). We also report the human performance for this task in terms of the above metrics computed between annotations from the two judges.

A trivial baseline – which concatenates tokens from the 'sent-to' and 'subject' fields and the commitment sentence – is included for comparison.

The best performance is obtained with Seq2Seq using copying mechanism. We observe our model to perform at par with human performance for writing To-Do items. Table 3 shows some examples of To-Do item generation from our best model.

6 Conclusions

In this work, we study the problem of automatic To-Do item generation from email context and meta-data to provide smart contextual assistance in email applications. To this end, we introduce a new task and dataset for action-focused text intelligence. We design a two stage framework with deep neural networks for task-focused text generation.

There are several directions for future work including better architecture design for utilizing structured meta-data and replacing the two-stage framework with a multi-task generation model that can jointly identify helpful context for the task and perform corresponding text generation.

References

Hosein Azarbonyad, Robert Sim, and Ryen W White. 2019. Domain adaptation for commitment detection in email. In *Proceedings of the twelfth ACM international conference on web search and data mining*, pages 672–680.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *ICLR*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Giuseppe Carenini, Raymond T Ng, and Xiaodong Zhou. 2007. Summarizing email conversations with clue words. In *Proceedings of the 16th international conference on World Wide Web*, pages 91–100. ACM.

Vitor R Carvalho and William W Cohen. 2005. On the collective classification of email speech acts. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 345–352. ACM.

Mia Xu Chen, Benjamin N. Lee, Gagan Bansal, Yuan Cao, Shuyuan Zhang, Justin Lu, Jackie Tsay, Yinan Wang, Andrew M. Dai, Zhifeng Chen, Timothy Sohn, and Yonghui Wu. 2019. Gmail smart compose: Real-time assisted writing. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19. ACM.

Simon Corston-Oliver, Eric Ringger, Michael Gamon, and Richard Campbell. 2004. Task-focused summarization of email. In *Text Summarization Branches Out*. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Mark Dredze, Hanna M Wallach, Danny Puller, and Fernando Pereira. 2008. Generating summary keywords for emails using topics. In *Proceedings of the 13th international conference on Intelligent user interfaces*, pages 199–206. ACM.

Tanya Feddern-Bekcan. 2008. Google calendar. *Journal of the Medical Library Association: JMLA*, 96(4):394.

Anjuli Kannan, Karol Kurach, Sujith Ravi, Tobias Kaufmann, Andrew Tomkins, Balint Miklos, Greg Corrado, Laszlo Lukacs, Marina Ganea, Peter Young, et al. 2016. Smart reply: Automated response suggestion for email. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 955–964. ACM.

Bryan Klimt and Yiming Yang. 2004. The enron corpus: A new dataset for email classification research. In *European Conference on Machine Learning*, pages 217–226. Springer.

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*, pages 984–992. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Chu-Cheng Lin, Dongyeop Kang, Michael Gamon, and Patrick Pantel. 2018. Actionable email intent modeling with reparametrized rnns. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Douglas Oard, William Webber, David Kirsch, and Sergey Golitsynskiy. 2015. Avocado research email collection. In *LDC2015T03. DVD. Philadelphia: Linguistic Data Consortium*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Sara Radicati and J Levenstein. 2015. Email statistics report, 2015-2019. *Radicati Group, Palo Alto, CA, USA, Tech. Rep*.

Owen Rambow, Lokesh Shrestha, John Chen, and Chirsty Lauridsen. 2004. Summarizing email threads. In *Proceedings of HLT-NAACL 2004: Short Papers*, HLT-NAACL-Short '04. Association for Computational Linguistics.

Simon Scerri, Gerhard Gossen, Brian Davis, and Siegfried Handschuh. 2010. Classifying action items for semantic email. In *LREC*.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

I Sutskever, O Vinyals, and QV Le. 2014. Sequence to sequence learning with neural networks. *Advances in NIPS*.

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.

Rui Zhang and Joel R. Tetreault. 2019. This email could save your life: Introducing the task of email subject line generation. In *ACL*.

A Appendix

A.1 Hyper-parameters

We now provide the hyper-parameters and training details for ease of reproducibility of our results. The encoder-decode architecture consists of LSTM units. The word embedding look-up matrix is initialized using Glove embeddings and then trained jointly to adapt to the structure of the problem. We found this step crucial for improved performance. Using random initialization or static Glove embeddings degraded performance.

We also experimented with using either a shared or a separate vocabulary for the encoder and decoder. A token was included in the vocabulary if it occurred at least 2 times in the training input/target. Separate vocabulary for source and target had better performance. Typically, source vocabulary had higher number of tokens than target. A shared dictionary led to increased number of parameters in the decoder and to subsequent over-fitting. The validation data was used for early stopping. The patience was decreased whenever either the validation token accuracy or perplexity failed to improve. We used the OpenNMT framework in PyTorch for all our Seq2Seq experiments.

Table 5 lists the hyper-parameters of the best performing model.

Hyper-parameter	Value
Rnn-type	LSTM
Rnn-size	256
# Layers	1
Word-embedding	100
Embedding init.	Glove
Batch size	64
Optimizer	Adagrad
Learning rate	0.15
Adagrad accumulator init.	0.1
Max. Gradient norm	2.0
Dropout	0.5
Attention dropout	0.5
Tokenizer	spacy
Vocabulary	Separate
Early Stopping (Patience)	5
Beam width	5

Table 5: Seq2Seq with copy mechanism : Hyper-parameters for the best model.

A.2 Illustrative Examples

In this Section, we provide further examples of the email threads along with the highlighted commitment sentence. Note that some of the emails have previous thread email present, and some do not have it. For each of these examples, we also provide the To-Do item written by the human judge (denoted as GOLD) and that predicted by our best model (denoted as PRED). As in the main text, the sentences have been paraphrased and names changed due to the data sensitivity of Avocado.

From: Beverly Evans *To:* Carlos Simmons *Subject:* Amazon.com update

Carlos,

I came to know today from John Carter than we received a PHP script that is not decoding the correct database. Can you check with them why they sent us the eCommerce PHP code when the loss of functionality was not out fault? I have registered the error log in the eCommerce section because the staff scientist from Amazon mentioned it in his email. He also said they have not been able to resolve the issue and surprisingly did not mention who we should contact next. (This email exchange was about a week ago when I had handed them the cloud expenditures.) Also, we need to generate a PHP example to replicate the error. Could you update me if the team is working on it?

Thanks, Beverly

From: Carlos Simmons *To:* Beverly Evans *Subject:* Amazon.com update

The PHP they shared with us is an example. eCommerce is not what they want us to resolve. I feel we should wait until their engineers test all possibilities. Joseph informed us that they need to test the database more carefully and figure out which PHP code to send to us and whether they want our feedback on the database. I am not sure why they sent me a 'relevant PHP example' - I thought there was the only file they sent us yesterday. I will forward that to you and Renata.

GOLD: Forward PHP example to Beverly and Renata.

PRED: Forward eCommerce PHP to Beverly.

Table 6: Illustrative Example 1

From: Kirstin Barnes *To:* Nannie Jacobs *Subject:* Ready for Product Launch

Nannie,

I am ready for the product launch. I need to include some of the enhancements in the presentation. I'll submit what is already completed and then do the remaining after the meeting..

Kirstin Barnes

Product Engineer AvocadoIT, Inc.

GOLD: Submit presentation with product enhancements.

PRED: Submit the enhancements for product launch.

Table 7: Illustrative Example 2

From: Rishabh Iyer *To:* R&D *Subject:* Software not ready yet for deployment

Hello,

Unlike our plan last month, the software is still not ready for deployment. The team put together some errors last week. We must plan to make it available latest by next week. I will keep you posted.

Thanks, Rishabh Iyer.

Software Engineer AvocadoIT Inc.

GOLD: Keep r&d posted about deployment of software.

PRED: Keep r&d posted about deployment.

Table 8: Illustrative Example 3

From: Justine Sparrow *To:* Roma Patterson *Subject:* 24x7 Helpline

Roma,

I will bring this up in the Staff meeing today. I'll let you know the outcome. Could you confirm if this is for a license agreement or a shared solution ?

Thanks, Justine.

GOLD: Let Roma know result.

PRED: Let Roma know about the license agreement.

Table 9: Illustrative Example 4

From: Rebecca Anderson *To:* Julia Roberts *Subject:* Run a bash script while synchronize Julia, When synchronizing is done, we want to run a bash script to delete old records on the machine and remove all activity logs. How can I do this ? What is the way to perform this operation ? Also, in the bash script, is there a way to sort the dates so that we can identify older activities ? Thanks, Rebecca.
From: Julia Roberts *To:* Rebecca Anderson *Subject:* Run a bash script while synchronize Rebecca, We had exactly the same feature to delete activities which you mentioned in our previous release. But we no longer have that in the new version due to resource constraints. I will take to John to review this again. Thanks, Julia.
GOLD: Talk to John to review bash script again.
PRED: Talk to John to review the activities.

Table 10: Illustrative Example 5

From: Ramesh Paul *To:* Gopal Majumdar *Subject:* Updates List for 3/11 Here's the update for this week. 1. The R&D team is working on a presentation for the knowledge tranfer for v5. It should be ready within next two weeks. 2. I have received their email, but need to review the ppt. 3. Did you want to know more about the new cloud feature for automatic version management ? Or was it a different feature ? 4. I am constantly working on this. 5. Didn't we discuss this point in our last email ? 6. We are making similar tests in the desktop for v5 before migrating to the cloud. We first have to make sure things work well for the desktop. I will send you more details soon. Did you get a chance to update your blog with information about these new features ? Thanks, Ramesh.
GOLD: Send Gopal more details about tests in the desktop for v5.
PRED: Send Gopal more details on presentation for the knowledge transfer.

Table 11: Illustrative Example 6

From: Lori Howard *To:* Karen James; Bruce Thomas; Steve Perry *Subject:* Room reservations Team, This needs to be done through a formal training session, but as of now let me point out some crucial points about room reservations. 1. In case you allocate a room for general meetings and administrative work, then make sure you book it for that month, but not for long periods of time. (Karen, can you check with Renata whether this is fulfilled for our meetings next week?) 2. In case of clients who do not need the entire month, make sure to reserve only for the particular month. If it exceeds that time, the system will authomatically resolve it and reserve it for next month. 3. For room reservation, either enter the number of hours required or the % of month, but not both. I would prefer precise hours. I will inform you when we can provide training, perhaps we can next week. Thanks, Lori.
GOLD: Let Karen know about the training provide for room reservations.
PRED: Let Karen know about room reservations.

Table 12: Illustrative Example 7

From: Matthew White *To:* Frank; Paul; Dennis *Subject:* Draft Agenda for Software Training Dear All, As discussed before, we have finally come to a concrete plan. I have attached the draft for your review. Please go over it and let me know asap your suggestions so that I can send them to the organizers. Please check the agenda and the names of trainees. I'll put together the Training plan and the overall 5-day agenda as soon as I can. Matthew.
GOLD: Put together the training plan and the overall day agenda of software training.
PRED: Put together the draft agenda for software training.

Table 13: Illustrative Example 8

From: Diana Wilson *To:* Alba Deacon *Subject:* DHL package from IBM Alba, I was able to track the package and as per the website it was in Sao Luis, Brazil at noon. I am not sure where it is, but it is Brazil so ... Send me an update if you receive it from them. I just tracked the package and as of 10:00am today it was in Toluca, Mexico. Where that is I have no idea but it is in Mexico so ... Let me know if you hear from them when they receive it. Thanks. Diana Wilson.
From: Alba Deacon *To:* Diana Wilson *Subject:* DHL package from IBM Thanks Diana. If I hear anything I'll let you know.. Alba.
GOLD: Let Diana know about DHL package from IBM.
PRED: Let Diana know about DHL package from IBM.

Table 14: Illustrative Example 9