# Smart To-Do : Automatic Generation of To-Do List from Emails


**Sudipto Mukherjee[†], Subhabrata Mukherjee[‡], Marcello Hasegawa[‡], Ahmed Awadallah[‡] and Ryen White[‡]**

[†]University of Washington, [‡]Microsoft Research, Redmond

{sudipm}@uw.edu, {submukhe, marcellh, hassanam, ryenw}@microsoft.com



## Abstract

Smart contextualized features in email service applications increase the ease with which people organize their folders, write their emails and respond to pending tasks. In this work, we explore an interesting application that allow users to diligently organize their tasks and schedules via Smart To-Do. 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 human-level performance for this given task. To the best of our knowledge, this is the first work to address composing smart To-Do items from emails.


## 1 Introduction

**Motivation:** Email is an integral part of communication in enterprise, academic and inter-personal settings (Radicati and Levenstein, 2015). With the growing number of users in email platforms for myriad applications such as online retail, instant messaging and calendar events (Feddern-Bekcan, 2008), service providers are constantly seeking to improve user experience. Smart Reply (Kannan et al., 2016) and Smart Compose (Chen et al., 2019) are two recent features that provide contextual assistance to users. Another effort in this dimension is for automated task management and scheduling. The recent Nudge feature in Gmail and Insights in Outlook are designed to remind users to follow-up on an email or pay attention to pending tasks.

**State-of-the-art and differences:** Summarization of email threads has been the focus of multiple research works in the past (Rambow et al., 2004; McKeown et al., 2007; Carenini et al., 2007; Dredze et al., 2008; Carenini et al., 2008; Zajic et al., 2008; Zhang and Tetreault, 2019). There has also been considerable research on identifying

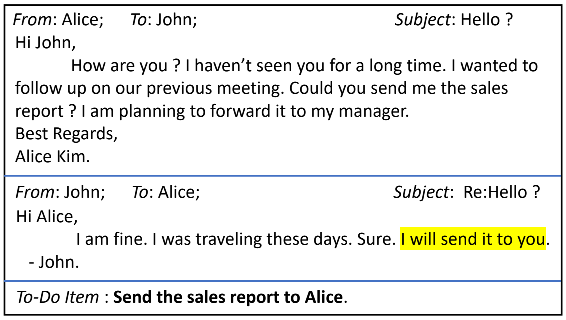


Figure 1: An illustration showing the email and a commitment sentence (in yellow) and the target To-Do item along with other email meta-data.

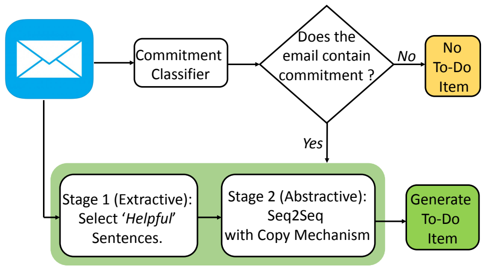


Figure 2: Smart To-Do flowchart.

speech acts or tasks in emails (Carvalho and Cohen, 2005; Lampert et al., 2010; Lin et al., 2018; Wang et al., 2019). However, there has been less focus on task-specific email summarization (Corston-Oliver et al., 2004) that is one of the main focus in this work. To-Do item generation is distinct from news headline, email subject line or email conversation summarization, since the focus of generation is a specific task outlined in the email.

**Overview of our work:** In this work we advance contextual intelligence for email communication by automatically generating To-Do items from email context and meta-data and assist users on following up on their promised actions (referred to as commitments). 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 provides contextual information about the query along with the entities (e.g., persons) associated with the task. We leverage existing work to identify the task sentence via a commitment classifier that detects action intents in the emails. Thereafter 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 leveraged to generate the To-Do item using a sequence-to-sequence method with deep neural networks. Figure 2 shows a schematic diagram of the process. Since there is no existing work or dataset on the same, we setup a task to collect annotated data for this task. Overall, our contributions can be summarized as:

- We create a new dataset for To-Do item generation from emails containing action items based on the publicly available email corpus *Avocado* (Douglas Oard and Golitsynskiy, 2015). [1]
- We develop a two-stage algorithm, based on unsupervised task-focused content selection and subsequent text generation leveraging contextual information and email meta-data.
- We perform experiments in this new dataset and show our model to perform at par with human judgments on multiple performance metrics.

## 2 Dataset Preparation

We leverage the Avocado dataset (Douglas Oard and Golitsynskiy, 2015)[2] containing an anonymized version of the Outlook mailbox for 279 employees with various meta information and $938,035$ emails overall.

### 2.1 Identifying Action Items in Emails

Emails contain various user intents including planning and scheduling for 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 some task, providing some information, keeping others informed about some topic, etc. We use the term *commitment* for such sentences in the email.

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. We used a recurrent neural network based on bidirectional GRU similar to the one in (Wang et al., 2019) (refer to Appendix for hyper-parameter settings). The classifier has a precision of $86\%$ and recall of $84\%$ on sentences in the Avocado corpus.

### 2.2 To-Do Item Annotations

We extracted $500k$ raw sentences from the emails and passed them through the commitment classifier. We selected those emails as candidates for the To-Do task where the predicted commitment probability of any of the sentences in the email is greater than $0.9$. A random subset of these are selected for annotation via crowdsourcing.

For each candidate email $e_c$ and the previous email in the thread $e_p$ (if present), we obtain metadata like '*From*', '*Sent-To*', '*Subject*' and '*Body*'. The commitment sentence in $e_c$ is highlighted and annotators are asked to write a To-Do item using all of the information in $e_c$ and $e_p$. Figure 1 shows an example of the annotation step. Each email is sent to two annotators.

### 2.3 Analysis of Crowd-sourced judgments

We now report some quantitative metrics obtained from the human judgments. Given the email metadata, the annotators were also asked whether the *subject* information was helpful in writing the To-Do Item. For $46.75\%$ of the candidate emails, both annotators agree that the subject was helpful. We obtained a total of 9349 instances with To-Do items available. These were split as $7349$ for training and $1000$ each for validation and testing. For each instance, we choose the annotation with fewer tokens as our ground-truth. To-Do items have a median token length of 9 and a mean of $9.71$ in this dataset.

## 3 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. The entire email contains generic sentences such as salutations, thanks and casual conversations not relevant for the task. The objective of the first stage is to select sentences containing informative concepts necessary to write the To-Do.

[1]We will release the dataset in accordance with LDC and Avocado policy.

[2]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.

### 3.1 Identifying Helpful Sentences for a 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 $e_c$ and previous email $e_p$ be $\{s_1, s_2, \ldots s_d\}$. 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 $\tau$ maximum frequency tokens from the *local vocabulary* consisting of tokens from all the email sentences, the commitment and the subject (i.e., $\{s_1, s_2, \ldots s_d\} \cup \mathcal{H} \cup \text{Subject}$). Tokens are lemmatized and English 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 $\tau$ tokens.

**Relevance score computation:** Task-specific relevance score $\Omega$ 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. $\Omega(s_i) = h(s_i)^T h(\mathcal{E})$.

The objective is to find helpful sentences for the task 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 provides useful information via the enriched context. Further, we experiment with three different embedding functions.

(1) Term-frequency (Tf) - The binarized term frequency vector based on tokens in the local vocabulary 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 - In contrast to FastText generating word embeddings agnostic of context, we leverage recent advances in contextualized representations from pre-trained language models like BERT (Devlin et al., 2019). We used the second last layer from BERT for the embeddings. We found this to perform better for our task than the last layer that was optimized for a different task-specific loss during BERT pre-training.

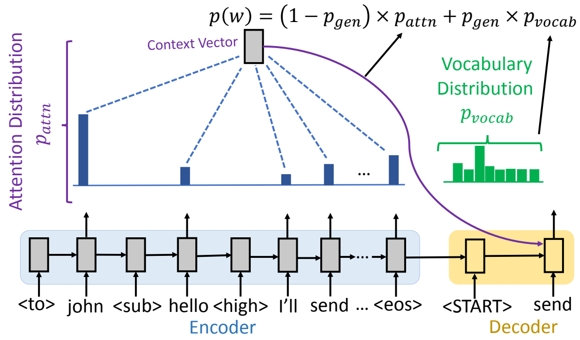


Figure 3: Seq2Seq with copy mechanism.

**Evaluation of unsupervised approaches:** Retrieval of at-least one helpful sentence is crucial to obtain the concepts required to write the To-Do item. Therefore, we evaluate our unsupervised 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 all the sentences 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.

Table 1 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.

| 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-trained) | 0.80 | 0.89 |

Table 1: Performance of unsupervised approaches in identifying helpful sentences for a task.

| 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 2: To-Do item generation results.

### 3.2 To-Do Item Generation

The generation phase of our task can be formulated as a 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 con-

| Label | Generated Text |
|---|---|
| GOLD | Let john know about the training provide for booking resources. |
| PRED | Let john know about booking resources. |
| GOLD | Put together the se plan and the overall day agenda of sales training. |
| PRED | Put together the draft agenda for sales training. |
| GOLD | Let elisabeth know about fedex package for hp. |
| PRED | Let elisabeth know about fedex package for hp. |
| GOLD | Let alex know result. |
| PRED | Let alex know about the license deal. |

Table 3: Generation examples (GOLD: manual annotation, PRED: machine-generated) with email context, meta-data in Appendix.

sists 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:

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

The input tokens $\{x_1, x_2, \dots x_T\}$ 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 $t$. This is the first baseline (referred as vanilla Seq2Seq) in our work.

As the second model, 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 $h_t$.

$$\mathrm{e}_{t,t'} = v^T tanh(W_h \cdot h_t + W_s \cdot s_{t'} + b)$$
$$a_{t,t'} = softmax(e_{t,t'})$$
$$h_t = \sum_{t'} a_{t,t'} \cdot h_{t'}$$

As our third 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 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.

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 copy, the vocabulary is extended as $\mathcal{V}' = \mathcal{V} \cup \{x_1, x_2, \dots x_T\}$. The model is trained end-to-end to maximize the log-likelihood of target words (To-Do items) given the email context.

We also modeled the problem with query-focused attention (referred as Seq2Seq BiFocal) having two encoders – one containing only tokens of the query and the other containing rest of the input context. We used a bifocal copy mechanism that can copy tokens from either of the encoders (refer to Appendix for details of the training and hyper-parameters of our architectures).

## 4 Experimental Results

We trained the above neural networks for To-Do generation on our annotated dataset. The median token length input to the encoder is 43. Table 2 shows the results. We report the performance metrics in terms of BLEU-4 (Papineni et al., 2002) and the F1-scores of Rouge-1, Rouge-2 and Rouge-L (Lin, 2004). We also report a trivial baseline, which simply concatenates tokens from the 'sent-to' and 'subject' fields and the commitment sentence. The best performance is obtained with Seq2Seq using copying mechanism. We observe the model to perform at par with human-level performance in this task. We found the query-focused bifocal copy mechanism to be slightly worse than using a single copy-mechanism resulting from the ambiguity on whether to copy from the query or the context at a given timestep. We show examples of generated To-Do items for the best model in Table 3.

## 5 Conclusions

In this work, we explore the problem of automatic To-Do item generation from email context and meta-data to provide smart contextual intelligence for email applications. To this end, we created a new dataset using crowdsourcing and designed a two stage framework with deep neural networks for task-focused text generation.

## References


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.

Giuseppe Carenini, Raymond T Ng, and Xiaodong Zhou. 2008. Summarizing emails with conversational cohesion and subjectivity. In *Proceedings of ACL-08: HLT*, pages 353–361.

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*.

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

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*.

Kathleen McKeown, Lokesh Shrestha, and Owen Rambow. 2007. Using question-answer pairs in extractive summarization of email conversations. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 542–550. Springer.

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.

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.

David M Zajic, Bonnie J Dorr, and Jimmy Lin. 2008. Single-document and multi-document summarization techniques for email threads using sentence compression. *Information Processing & Management*, 44(4):1600–1610.

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

## A 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.

Table 4 lists the hyper-parameters of the best performing model.

## B 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).

| **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 4: Seq2Seq with copy mechanism : Hyper-parameters for the best model.

| |
|---|
| *From:* Bryan Wiens *To:* Meshele Ko *Subject:* AC.COM update<br>Meshele,<br>I was just informed by John Schemena that we received an XML datafeed for funcitonality that is not in the DDS. Could you please find out why they are sending us the mCommerce XML feed when it was not part of our Requirements call? I had included a placeholder for the mCommerce section on the menu because Meg from AC had made reference to it once, but she said that she is not in control of that portion and never asked us to get in touch with anyone for that functionality. (This call took place about a week before you and I called her together for the requirements.) Also, we still need to get a relevant XML example. Do you know when this will be comming?<br>Thanks, Bryan |
| *From:* Meshele Ko *To:* Bryan Wiens *Subject:* AC.COM update<br>The xml they shared with us was a sample. mcommerce may not be what they ask us to include. i think we need to hold until they do their part. Bonnie committed to us that they need to take a look at the DDS and figure out what xml to send to us and if they to give us feed back on the DDS. I am confused about getting a 'relevant XML example' - I thought that is what they sent us yesterday. I will forward that to you and elba. |
| GOLD: Forward xml example to bryan and elba. |
| PRED: Forward mcommerce xml to bryan. |

Table 5: Illustrative Example 1

| |
|---|
| *From:* Ricardo Garcia *To:* Dan Baca *Subject:* Ready for Demo<br>Dan,<br>Dan, I am ready for the demo. I am still working on documenting the product enhancements. I'll submit what I have when you call me and then submit the rest after the meeting..<br>Ricardo Garcia<br>Sales Engineer AvocadoIT, Inc. |
| GOLD: Submit documenting the product enhancements. |
| PRED: Submit the product enhancements for demo. |

Table 6: Illustrative Example 2

| |
|---|
| *From:* Srikanth Raghavan *To:* R&D *Subject:* Build is not available yet for testing<br>Hello,<br>The build is not available as originally planned for this morning. There were some build error's last night. We will plan on making it available later this afternoon.I will keep you posted.<br>Thanks, Srik Raghavan.<br>Server Engineering AvocadoIT Inc. |
| GOLD: keep r&d posted about testing for building. |
| PRED: keep r&d posted about testing. |

Table 7: Illustrative Example 3

| |
|---|
| *From:* Lisa Quaglietti *To:* Alex Tran *Subject:* 24x7 Support<br>Alex,<br>I'm going to bring this up in eStaff today.I'll let you know the outcome. Can you tell me, is this for a license deal or a hosted solution?<br>Thanks, Lisa. |
| GOLD: Let alex know result. |
| PRED: Let alex know about the license deal. |

Table 8: Illustrative Example 4

| |
|---|
| *From:* Joseph Raymond *To:*support@avocadoit.com *Subject:* Err 4<br>Hi,<br>I am getting a periodic err 4 from the quotes request using sprint pcs touchpoint phone. It may be an xml err on our side, but can you take a look.<br>Thanks, Joe Raymond. |
| *From:* Brian Robinson *To:* Joseph Raymond *Subject:* Err 4<br>Good Morning Joe, I'll take a look at it and get back to you. |
| GOLD: Take a look at Err 4 and get back to Joseph. |
| PRED: Take a look at periodic and get back to joseph |

Table 9: Illustrative Example 5

| |
|---|
| *From:* Shivaprasad Madishetti *To:*Lisa Chui *Subject:* Call an act file while synchronize<br>Lisa,<br>After synchronizing, we want to call an act file and delete all the activities which are more than one month old on the device. How to do this. How can we achieve this functionality. And also in the act file, can we specify '¡' or '¿' in Xpath to identify the activities older than some date.<br>Thanks, Shiva. |
| *From:* Lisa Chui *To:* Shivaprasad Madishetti *Subject:* Call an act file while synchronize<br>Shiva,<br>Your description is exactly what we have discussed in 4.6 release. However we have dropped that feature due to resource constrains. I will talk to Ravi to review this again..<br>Thanks, Lisa. |
| GOLD: Talk to ravi to review act file again. |
| PRED: Talk to ravi to review the activities |

Table 10: Illustrative Example 6

| |
|---|
| *From:* Helen Spade *To:* Emma Fowler; Dan Baca; Toni Oliveria *Subject:* Thanks<br>Thanks for your help in pulling together this draft presentation for the Sun eIntegrator conference. Attached is the presentation. Please let me know of any changes you want made to the next rev. I sent to Sun already and am asking for their changes. I will keep you posted.<br>Thanks, Helen. |
| GOLD: Keep emma posted about draft presentation for sun eIntegrator conference. |
| PRED: Keep emma posted about sun eintegrator conference |

Table 11: Illustrative Example 7

| |
|---|
| *From:* Darshan Patel *To:* Divakar Tantravahi *Subject:* Issues list for 7/7<br>I have a few answers. 1. The R&D team is working on a presentation for the India Operations for the knowledge transfer for v2.1. They plan to have it ready for week of 7/17. 2. got the email, need to review the situation. 3. ? Are you looking for the URL which represents the final location once we submit the application to CO-LO and the customer? Or is this something different? 4. Always working on this one. 5. Is this different from the dedicated line we discussed before? 6. We are trying to decide that one here also. We have someone that is testing all our applications on v2.1 to make sure things still work. After that we will have a better idea if we should switch over or wait. I'll send you more details soon. Were you able to write a brief summary of the India operations for the Newsletter yet?<br>Thanks, Darshan. |
| GOLD: Send divakar more details about testing application on v2. |
| PRED: Send divakar more details on presentation for the knowledge transfer. |

Table 12: Illustrative Example 8

| |
|---|
| *From:* Lisa Quaglietti *To:* John Schemena; Divakar Tantravahi=; Richard Yoza *Subject:* Booking resources<br>Team,<br>We'll need to provide further training on this, but for now you need to know a couple of things about booking resources. 1. If you are going to allocate internal time to a resource for general meetings and administrative work, you must book it by month, not for an extended period of time. (John, can you work with Elisabeth to correct this for the AEs you put in last week?) 2. For customer projects being booked where the project is less than 100% of an individuals time, you will need to book the time by month. If you fail to do this the system will spread the time inaccurately. If you are booking a resource 100% for a period of time and it spreads over a several month period, you don't need to book the time by month. 3. When booking resources, you need to use either hours or %, but not both. My preference would be that we try to use hours where possible. We'll let you know what day we can provide training, we'll shoot for this week.<br>Thanks, Lisa. |
| GOLD: Let john know about the training provide for booking resources |
| PRED: Let john know about booking resources. |

Table 13: Illustrative Example 9

| |
|---|
| *From:* Chris Longstaffe *To:* Don; Glenn; Alex; Matt *Subject:* Draft Agenda for Sales Training<br>Gentlemen,<br>Following our various discussions here's the current plan. Please review the attached draft and let me have any comments back asap so that we can distribute this to the team. Please check both the agenda and the potential attendees. I'll put together the SE plan and the overall 4-day agenda as soon as I can.<br>Chris. |
| GOLD: Put together the se plan and the overall day agenda of sales training. |
| PRED: Put together the draft agenda for sales training. |

Table 14: Illustrative Example 10

| |
|---|
| *From:* Elisabeth Whaley *To:* Robert Moore *Subject:* Fedex package for HP<br>Robert,<br>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.<br>Thanks. Elisabeth V. Whaley. |
| *From:* Robert Moore *To:* Elisabeth Whaley *Subject:* Fedex package for HP<br>Thanks Elisabeth. If I hear anything I'll let you know..<br>Robert. |
| GOLD: Let elisabeth know about fedex package for hp. |
| PRED: Let elisabeth know about fedex package for hp. |

Table 15: Illustrative Example 11

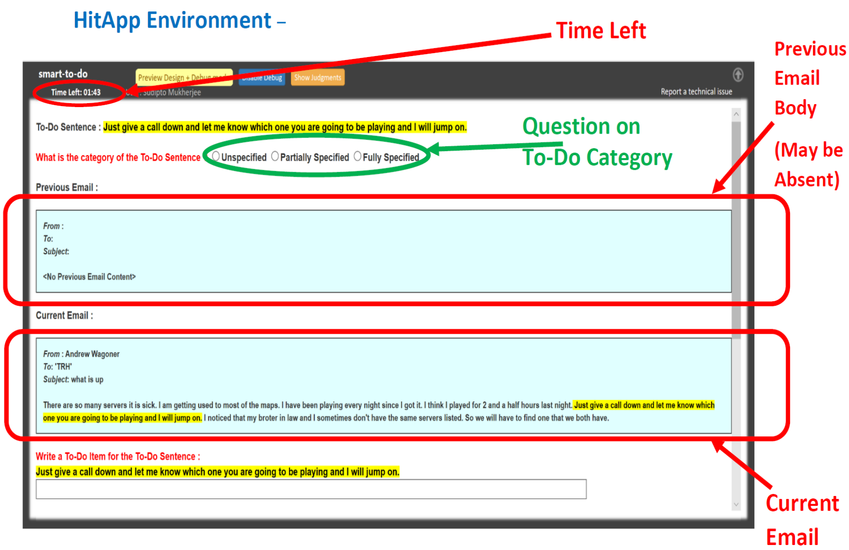


Figure 4: HitApp Environment