Automatic Rephrasing of Transcripts-based Action Items

Amir DN Cohen[1,2], Amir Kantor[1], Sagi Hilleli[1] and Eyal Kolman[1]
Microsoft[1]
Department of Computer Science, Bar-Ilan University[2]
amirdnc@gmail.com,{amir.kantor,sagih,eyalko}@microsoft.com

Abstract

The automated *transcription* of *spoken language*, and meetings, in particular, is becoming more widespread as *automatic speech recognition* systems are becoming more accurate. This trend has significantly accelerated since the outbreak of the COVID-19 pandemic, which led to a major increase in the number of online meetings. However, the transcription of spoken language has not received much attention from the NLP community compared to documents and other forms of *written language*. In this paper, we study a variation of the summarization problem over the transcription of spoken language: given a transcribed meeting, and an *action item* (i.e., a commitment or request to perform a task), our goal is to generate a coherent and self-contained *rephrasing* of the action item. To this end, we compiled a novel dataset of annotated meeting transcripts, including human rephrasing of action items. We use state-of-the-art supervised text generation techniques and establish a strong baseline based on BART and UniLM (two pretrained transformer models). Due to the nature of natural speech, language is often broken and incomplete and the task is shown to be harder than an analogous task over email data. Particularly, we show that the baseline models can be greatly improved once models are provided with additional information. We compare two approaches: one incorporating features extracted by coreference-resolution. Additional annotations are used to train an auxiliary model to detect the relevant context in the text. Based on the systematic human evaluation, our best models exhibit near-human-level rephrasing capability on a constrained subset of the problem.

1 Introduction

Most of *natural language processing (NLP)* research focuses on written language, such as emails or Web pages, and less on an increasingly large body of spoken language converted to text via *automatic speech recognition (ASR)*. Particularly, today, more and more meetings are conducted online, especially since the COVID-19 social distancing constraints. Online meetings may be transcribed upon request, generating a huge amount of spoken language text.[1]

Spoken language has different characteristics from written language, and, from our experience, it is typically vaguer and harder to understand: sentences tend to be broken, less orderly, incomplete (relying on subtext), and prone to speech-to-text transformation errors (see Section 5).

Action items (AIs) are a common and particularly important part of workplace meetings. An AI is a commitment or a request to perform a certain task by any of the parties involved. For example,

> *"I will send you the file later today."*

AIs occur naturally during conversations, but are not always clear without relevant *context*. For example, the AI *"I will do it"* contains a commitment to act, but the nature of this action is unclear. Nevertheless, the context of the AI might make it clearer; e.g., if the AI is preceded by a sentence such as *"Can you prepare a presentation for Thursday?"*. In the context of written communication, action items have been researched in a variety of ways, such as AI detection (Bennett and Carbonell, 2007, 2005), summarization and rephrasing of AIs in emails (Mukherjee et al., 2020; Rambow et al., 2004), and more. Verbal communication, in contrast, has not received much attention.

In this paper, we focus on the task of *rephrasing* an action item from a transcribed meeting into a coherent and self-contained utterance. Such an utterance may be also referred to as a paraphrase,

[1] For example, Microsoft Teams, a popular online communication platforms, has reached 115 Million daily users on October 2020.

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 2862–2873
August 1–6, 2021. ©2021 Association for Computational Linguistics

or a summary. Conceptually, the rephrasing task may be split into two sub-tasks: one is *locating* those parts in the context that are relevant to the AI; the other is *forming* a coherent and self-contained AI sentence from the original AI and the relevant context. While even "off the shelf" summarization models (e.g., HIBERT (Zhang et al., 2020)) may do well in forming an action item sentence, manual observation of the data showed that such models often fail to locate relevant context in the transcript. As we show in Section 6, the same applies to models trained specifically for the rephrasing task too. To overcome this, we introduce *"hints"* – additional annotations that are added to the rephrasing models' input to help the model better locate relevant spans of text (see Figure 1). We use two kinds of hints: *coreference hints*, which are obtained from a pre-trained model for the coreference resolution task, and *context hints* that are generated by a model trained specifically for the task based on our train data.

Our contribution may be summarized as follows:

1. We created a new dataset (coined *AIR*) including AIs extracted from transcribed meetings, with human annotations of context and AI rephrasing.

2. We show that models intended for AI rephrasing in email perform considerably worse on transcript data. Accordingly, we train transformer-based models for AI rephrasing – both on email data, reporting new state-of-the-art performance, as well as on transcript data – so as to form a baseline for the task.

3. We show that the baseline can be greatly improved by adding "hints" to the model's input. This results in near-human-level performance on a constrained subset of the problem. We believe that this approach may apply to various other problems.

We support our claims by extensive experimentation and evaluation, including independent human evaluation.

2 Related work

While AI extraction and summarization of emails have been researched extensively (Lin et al., 2018; Mukherjee et al., 2020; Rambow et al., 2004; Scerri et al., 2010), meeting transcripts have not received as much attention from the community. Closest to our work is the work of Mukherjee et al (Mukherjee et al., 2020), which focuses on the extraction and rephrasing of AIs over emails. While their goal is very similar to ours, rephrasing transcripts is very different from rephrasing emails, as we show in Section 5.

A: At the b- furthermore , I told Jerry that I was gonna finish it before he got back. So.
C: That's approaching. He's coming back when ? next -
A: I think - we think we'll see him definitely on Tuesday for the next - Or, no, wait. The meetings are on Thursday .
A: Well, we'll see him next week .
A: I was thinking about that. I think I will try to work on the SmartKom stuff and I'll if I can finish it today , ▲ I'll help ▼ you with that tomorrow.
A: if you work on it?
A: I don't have a problem with us working on it though? So.

(a) Predicted context annotations highlighted

A: At the b- furthermore , [c:0 I] told Jerry that [c:0 I] was gonna finish [c:4 it] before he got back. So.
C: That's approaching . He's coming back when ? next -
A: [c:0 I] think - we think we'll see him definitely on Tuesday for the next - Or, no, wait. The meetings are on Thursday.
A: Well, we'll [c:3 see] him next week.
A: [c:0 I] was thinking about [c:3 that]. [c:0 I] think [c:0 I] will try to work on [c:4 the SmartKom stuff] and [c:0 I]'ll if [c:0 I] can finish [c:4 it] today , ▲ [c:0 I]'ll help ▼ [c:0 you] with [c:4 that] tomorro ,
A: if [c:0 you] work on [c:4 it]?
A: [c:0 I] don't have a problem with us working on [c:4 it] though ? So.

(b) Predicted coreference annotations in square brackets (including cluster indices c:`<index>`)

Figure 1: Meeting transcript from the test set annotated by (a) context model; (b) coreference model. AI appears between solid triangles. Predicted rephrasing—**baseline:** *"Speaker A will try to work on SmartKom stuff"*; **context:** *"Speaker A will help Speaker C with SmartKom stuff"*; **coreference:** *"Speaker A will try to work on SmartKom stuff and if he can finish it he will"*; **human:** *"Speaker A will meet Jerry next week and will try to work on the SmartKom stuff"*.

Einolghozati et al. (2020) applied a pre-trained BART with a copy mechanism for the task of rephrasing virtual assistance messages. However, they are focused on style adaptation and personal pronouns modification while we focus on context-based enrichment.

Meeting summarization has been explored in the past (Oya et al., 2014; Garg et al., 2009). However, these works focus on full meeting transcript summarization, where we focus on extracting and rephrasing information specific to a given AI. In this perspective, our work can be viewed as a variation of the known *query based summarization* problem (Rahman and Borah, 2016), where given a document and a query, the algorithm goal is to extract information regarding the query from the document. While this topic has been studied in the past (Saggion et al., 2003; Bosma, 2003; Nema et al., 2017), to the best of our knowledge, there has been no work that relates to meeting transcripts, especially in the context of AIs.

3 The AIR Dataset

We create AIR **A**ction **I**tem **R**ephrasing, a new dataset focused on professional meetings AI rephrasing. The dataset is composed of instances, where each instance contains ten utterances, where the 8th utterance includes an AI. The labels are a human-produced rephrasing of the AI. The annotation process was split into three parts; acquiring raw data from multiple sources, AI extraction, and AI rephrasing. We now describe each of these steps.

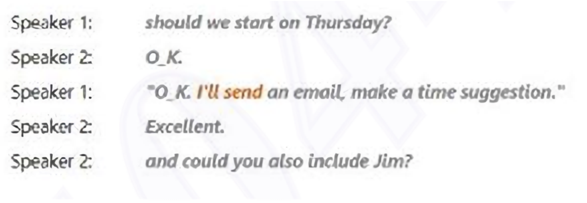
3.1 Dataset Construction

Several datasets of manually transcribed records of meetings were used to accumulate action items and generate rephrasing. The result is a diverse dataset, containing a collection of different meeting types, such as software development, product design, financial, and board meetings[2].

ICSI meeting corpus (Janin et al., 2003) contains 75 meetings recorded in a conference room at the International Computer Science Institute in Berkeley.

Augmented Multi-party Interaction (AMI) meeting corpus (Carletta et al., 2006) is a multimodal dataset consisting of 100 recording hours of 154 meetings, and their manually annotated transcripts. Some of the meetings are naturally occurring, and some are elicited, particularly using a scenario in which the participants play different roles in a design team, taking a design project from kick-off to completion over the course of a day.

Board Meetings (LSC) is an open to public board meetings transcripts of the legal services corporation (LSC) and other transcribed board meetings that were extracted from available public resources.



Speaker 1: *should we start on Thursday?*
Speaker 2: *O_K.*
Speaker 1: *"O_K. I'll send an email, make a time suggestion."*
Speaker 2: *Excellent.*
Speaker 2: *and could you also include Jim?*

Figure 2: The UI used by the human annotators for AI detection.

Internal Dataset (ID) that contains internal ***[3] manually transcribed meetings which mostly revolve around software development topics.

Parts of these datasets contain sensitive information, so while datasets will be made publicly available in the future, some parts of the data cannot be shared.

Table 1 describes relevant AIR statistics. Note that to better compare the two AIR versions, the dev and test set are identical between the public and restricted versions. Both development and test sets were composed of ISCI dataset. we used ICSI for the test and development set because ISCI samples contained the highest intra-judgment score as discussed in Appendix A.

3.2 Action Items Detection

Action Items are rare in conversations and are found in roughly 1% of sentences. To reduce annotation costs, we wish to increase the percentage of AIs in the data. To this goal, the transcripts are filtered by a pre-classifier – a list of regular expressions. The pre-classifier filters out 93% of the sentences, with a precision of 17% and recall of 90% over Action Items.

AI candidates that passed the pre-classifier were labeled using human annotators. The action item annotation task was composed of five sequential utterances. The sentence that includes the AI candidate was the third sentence and two previous and following utterances were shown as context, as seen in Figure 2. Each annotator was asked if the third utterance contains an AI or not.

As preparation for the annotation task, each judge reviewed the annotation guideline and was required to successfully pass a test of ten samples before granted access to the data. The candidate sentences were tagged by five annotators. If the agreement between judges was lower than 80%, four more annotators were added. Each AI candidate was labeled as Action Item or not, according

[2] Each of the datasets was adjusted slightly to make relatively uniform samples. e.g. all speakers roles were converted to the form ”speaker_X, where X is a running number

[3] Organization name is omitted to preserve anonymity

Name	# of meetings	Average # of utterances	# of AIs	# of rephrasing
ID	594	374	5356	14820
ICSI	75	1428	722	2487
AMI	145	820	461	1445
Board Meetings	206	787	1283	3286
Overall	1020	598	7957	22038
Overall public	426	911	2519	7218

Table 1: AIR Source datasets statistics including the number of meetings, number of utterances per meeting, number of AIs extracted from the full dataset, and the number of rephrasing.

to a majority vote of the annotators.

3.3 Action Items Rephrasing

Action Items were rephrased by human annotators. Similar to the AI detection stage, the datasets were divided into samples that contain ten utterances, seven utterances before the AI, and two utterances after. Preliminary analysis showed an accelerated decline in context relevance when moving away from the action item, with less than 1% contribution to the seventh sentence before the AI. This justified the decision to present no more than seven pre-AI sentences. The distribution of context over the seven pre-AI and two post-AI utterances is shown in Figure 3

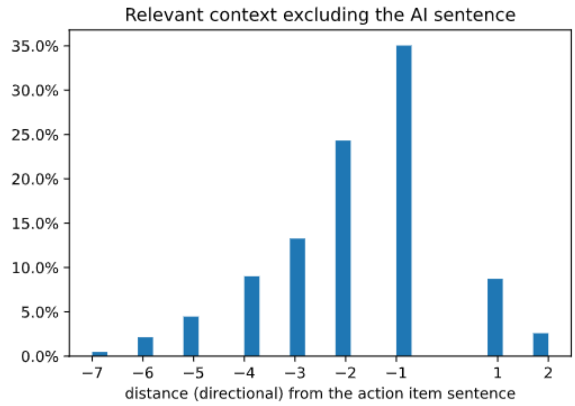


Figure 3: Percentage of the instances containing relevant context relative to the AI sentence.

While this truncation might make some AIs unclear due to lack of relevant content, this is a good trade-off between loss of information and annotation efficiency and accuracy. Human annotators were asked to write a self-explanatory sentence in their own words, based on the Action Items and their surrounding context. In text rephrasing, like other text generation tasks, there is not a single correct answer. Therefore, each sample was rephrased up to six times[4]. Multiple rephrasing per Action Item also helped us to assess the quality of the rephrasing. The overall number of AIs and Rephrasing ,train and test sets for the public dataset are described in table 2[5]. All samples in the test and validation sets were chosen from the ISCI dataset for uniformity (in both public and private variations).

Set	# of AIs	# of rephrasing
Train	2219	6318
Validation	150	450
Test	150	450

Table 2: Number of AIs and Rephrasings in train and test sets for the public dataset.

4 Model and Hints

In this section we show how to fine-tune a pre-trained model using the AIR data set. Additionally, we show how to add hints – extra annotations that improve the model ability to find relevant context, and generate more accurate and self-contained rephrasing. Hints and several versions of model rephrasing on the test set appear in Figure 1.

Base model The base model is based on BART (Lewis et al., 2020), a transformer based model (Vaswani et al., 2017), that was pre-trained by noising the input text and guiding the model to output, a de-noised version of the input. The BART model seems suitable for the rephrasing task. In this task, similar to the BERT pre-training task, the output

[4] Initially each sample was rephrased multiple times to check similarity of rephrasing and agreement. Later rehearsing was tagged between one to three times.

[5] Test and validation set were contracted as described in Appendix A

is a better version of the input; the ambiguity is cleaned from the AI and the output is a comprehensible and full version of the input. The de-noising ability of BART can also assist in overcoming noise of ASR systems or the speaker's partial and incoherent sentences. Our experiments in Section 6.2.2 further support this claim.

The model input is inspired by (Mukherjee et al., 2020). The input includes the sentence with AI, the previous seven sentences and the following two sentences. For each utterance in the input, we add the speaker name, and OOV markers to indicate the speaker's name and the AI, For example:

> <speaker>John</speaker><AI>I will send </AI>the presentation tomorrow.

The model is trained using teacher forcing and cross entropy loss over the predicted token.

4.1 Hints

One of the shortcomings of the base model is its limited ability to correctly identify relevant context with regard to the AI. Therefore, the rephrased AIs are often inaccurate and include irrelevant information from unrelated parts of the input text. To mitigate this, we propose two improvements to the base model.

Coreference model Given an input text, the task of *coreference resolution (CR)* aims to cluster entities that appear in different parts of the input text but refer to the same entity. This task has been thoroughly researched (Sukthanker et al., 2020; Soon et al., 2001), and in recent years gained a boost in performance, thanks to neural architectures (Xu and Choi, 2020; Joshi et al., 2020; Meged et al., 2020; Caciularu et al., 2021; Cattan et al., 2020). We hypothesize that CR models can capture semantic relations that the base model will miss because they are trained specifically for this task using vast amounts of data.

We use the CR model from Allennlp (Gardner et al., 2019), which uses SpanBERT for contextualized embedding (Joshi et al., 2020) and (Lee et al., 2017) method for CR. This model was trained on the OntoNotes dataset (Hovy et al., 2006).

Using the coreference model, we add hints to the text (see Figure 1b), identifying coreference clusters within the text. Text embedding includes cluster-marks appearing in square brackets around each of the cluster spans. For example:

> *"[c:0 Jon] works at the [c:1 cinema] , [c:0 he] loves working [c:1 there] ."*

We only mark clusters that have at least one instance in the AI utterance. This annotated text is input to the rephrasing model (both at the training phase and inference phrase). Note that training is applied only to the rephrasing model, whereas the CR model's weights are held constant.

Context detection model Another approach to improve the rephrasing model's ability to detect relevant spans is to directly train a model to detect them. To achieve this, we ask annotators to mark spans of text that are relevant to the AI. This data is used to train a *context detection model*. Spans of relevant text are transformed into binary token labels ('relevant' or 'irrelevant'). Accordingly, we train a token-classification model, based on the RoBERTa (Liu et al., 2019) pretrained transformer encoder, which is added to a fully-connected layer to perform per-token classification, given RoBERTa's output representation of each token. The model is fine-tuned end-to-end over the collected token annotations.

We found that the collected context information suffers from a low agreement between the judges. We use Kripendorff's alpha (Hayes and Krippendorff, 2007; Artstein and Poesio, 2008) coefficient to measure the judges (dis)agreement.We measure both an alpha score for the entire annotation task, as well as the pairwise agreement between the judges – both exhibiting values on the order of 0.4, which is considerably low. One reason for a low agreement may be an incomplete or incorrect understanding of the text: annotators are detached from the larger context of the meeting and from the subject matter; oral communication tends to be implicit and relies on pre-understanding; spoken sentences tend to be broken and less organized. Another reason might be the task itself, whose definition inevitably bears some level of ambiguity and arbitrariness.

Low agreement between annotators does necessarily undermine machine learning in its attempt to generalize from the train set (see Sect. 4.1.4 in (Artstein and Poesio, 2008), (Reidsma and Carletta, 2008)). It is important, however, to account for it when *measuring* classification performance on the test set. Each test instance for the context detection model consists of a transcript snippet containing a set of tokens for binary classification, and a human annotation of the tokens as relevant or irrelevant. Per test instance, we measure: (a) token

Model	Rouge-1	Rouge-2
SmartToDo (BiLSTM)	0.6	0.41
Human annotator	0.6	0.37
BART	0.63	0.43
BART (only subject & AI)	0.628	0.43
BART (No subject)	0.58	0.39
BART (AI only)	0.56	0.33

Table 3: Different variations of the email AI dataset.

ranking by the model (in accordance to the model's output scores) is evaluated by *average precision (AP)*; (b) F1-score of model's predictions. To aggregate on the test set, we have taken the mean of each value, resulting in *mean average precision (MAP)* and *mean F1-score (MF1)*.[6]

In order to assess the judges' disagreement on the test set, for each of the test instances, we collected 3 to 6 annotations. Per test instance, the optimal AP is obtained when ordering the token according to their ”soft” label – mean of judges' scores (0 or 1).[7] We denote by oMAP the mean of optimal AP over all test instances. Similarly, oMF1 is defined as the mean of instances' F1-scores for judges' majority prediction per token.

The context model's *normalized MAP* score (namely, $\frac{\text{MAP}}{\text{oMAP}}$) and *normalized MF1* score (namely, $\frac{\text{MF1}}{\text{oMF1}}$) are both 0.82. Similarly to the CR model, the context detection is used to add hints to the rephrasing models' input (see Figure 1a). Training the rephrasing model designates a second usage of the corpus, now with human rephrasing annotations.[8] Despite a relatively low agreement between the judges, context annotations significantly improve the rephrasing model's performance.

5 Comparison of Emails and Transcripts

AI rephrasing was applied over emails in (Mukherjee et al., 2020). We evaluate this model over meeting transcripts and show that despite its success over emails, it is less suitable for transcripts. At a first glance, email and meeting transcript AI rephrasing might seem similar. In this section, we challenge this assumption by highlighting three key differences.

Email subject We use the dataset from (Mukherjee et al., 2020) to evaluate performance over emails compared to transcript data. Their data is based on the Avocado dataset (Douglas Oard, William Webber, David A. Kirsch, 2015), where each instance contains a pair of emails (an email with an AI and the previous email in the correspondence), and a rephrasing of the AI. To build a rephrasing model, the authors used the following approach: 1. Chose relevant sentences from each email by similarity to the AI sentence. 2. Create an input that contains the chosen sentences, the mails authors, and the mails subjects, tagging each part of the data with dedicated markers. 3. Learn a model using a BiLSTM with copy mechanism (Zeng et al., 2016).

We test a number of variations based on this approach. 1. We replace the BiLSTM with BART (BART); 2. Similar to (1), but we omit the email's subject from the input (No subject); 3. Similar to (1), but remove all the email's body text besides the AI sentence (only subject & AI); 4. Leaving only the AI sentence (AI only). The results are presented in Table 3. The BART base model unsurprisingly outperforms the BiLSTM model. But surprisingly, a model that is exposed to the Subject alone performs almost as well as the full model. Additionally, most of the improvement compared to the base (AI only) model comes from adding the email's subject. This means that the body of the emails play a minimal role in the email's rephrasing. Unfortunately, meeting transcripts do not have an equivalent to a subject, which forces the model to rely solely on the transcript text.

Number of utterances While two emails often supply enough context for rephrasing, transcript samples contains ten utterances, which require the model to “find” the right spans from a relatively large pool of text.

Malformed language In contrast to written text, spoken language is less formal. This results in people making grammatical mistakes like stopping at the middle of a sentence (”I will take the... yes, that's right” or repeating words (”I... ah... I... think that it's OK”). This new grammar is very different from the pretrained text most pretrained models were trained on.

[6] Due to the nature of the problem, there are considerably more negatives than positives, and thus we choose two metrics that embody the precision/recall tradeoff, which is indifferent to true negative predictions, rather than the true-positive/true-negative tradeoff.

[7] AP calculation takes token-level steps, updating precision and recall according to *all* judges' annotations at once.

[8] In the training phase, as in the inference phase, the context model's predictions are used, rather than the judges' annotations.

Model	ROUGE-1	ROUGE-2	BLEU
Human	0.519	0.305	0.737
Base model	**0.645**	**0.459**	**0.763**

Table 4: ROUGE and BLEU score of the base model and human rephrasing.

6 Experiments

We evaluate our various models compared to a human annotator, and between themselves, in two ways. The first is automatic n-gram based metrics and the second is direct human evaluation. All models were trained using the Huggingface framework(Wolf et al., 2019) with the following configuration: **batch size:** 8, **lr:** $3 \cdot 10^{-5}$, **epocs:** 5 (20 with public dataset).

6.1 N-gram Based Metrics

There are a variety of n-gram based methods that evaluate the quality of text generation tasks (Lin, 2004; Banerjee and Lavie, 2005; Papineni et al., 2001). In this work, we use the ROUGE-1 and ROUGE-2 (Lin, 2004) as it is a widely used measure for summarization evaluation.

We evaluate the base model vs. a gold rephrasing produced by a human annotator. We also include another human rephrasing that represents human evaluation level. The results are shown in Table 4. Even the base model suppresses the human-produced rephrasing and achieves a higher ROUGE score. While this result looks impressive, we claim it simply highlights the limitation of ROUGE, and other n-gram based metrics to evaluate the rephrasing quality (see (Mathur et al., 2020)). We provide the following rational:

Frequent words Not all words have the same value when evaluating the quality of a sentence. For example, entity mentions and verbs are more relevant to the sentence meaning than stop words, but n-grams models give each word an equal weight for the overall score. While human annotators are good at finding rare words that convey meaning, automatic models are good at using very frequent words. Using these words increase the model's score, while not contributing to the quality of the rephrasing.

Lacking of ”true” gold samples Typically, machine-learning algorithms are evaluated based on a gold standard that is generated by humans. It is also typical for ground truth to have some error rate, and there are techniques to reduce the probability of error, e.g., taking a majority vote. In contrast, in the case of text generation, there are usually many results (sentences) that can be considered 'good', having very different wording. BLEU (Papineni et al., 2001) addresses this difficulty by comparing the generated text to a number of different human-generated labels. While this somewhat reduces the chance that a good text generation will get a low score, it does not eliminate it.

Models might outperform humans as we show in Section 6.2.1, our models might achieve higher quality than the human-produced rephrasing. N-gram based models treat each deviation from the human model as an error, although it might achieve a better paraphrasing[9].

For these reasons, we turn to human evaluation as the main models' evaluation method.

6.2 Human Evaluation

We evaluate our models by using the following procedure - Each instance contains the model input (transcript + speakers + AI) and three[10] different rephrasings. The judges were instructed to assign each rephrasing a score from the set {1,2,3}, while considering these questions (arranged by importance):

1. Does the action described in the rephrasing accurately describe the action in the context + AI?
2. Does the rephrasing contain all the details that are described in the context + AI?
3. Is the rephrasing grammatically correct?
4. How easy was it to understand the rephrasing?

The judges UI is shown in Figure 5 on Appendix B.

6.2.1 Main Results

Using the AIR-internal dataset, we compare the three models with each other and with a human-produced rephrasing. We applied two evaluations. First, we compare the human-produced rephrasing, the base model, and the coreference model. The

[9] This indicates that the automatic metrics might underestimate automatic models.

[10] We use three models per instance (instead of four) to reduce the load from the judges.

Models	Result
Human/Coref	**56.0% / 44.0%**
Human/Base	**68.2% / 31.8%**
Base/Coref	**39.4% / 60.6%**
Context/Coref	**59.1% / 40.9%**
Human/Context	50.9% / 49.1%

Table 5: Comparison of the base, coreference, and context models. Bold results are significant.

Models	Result
Base/Coref	**43.4% / 56.6%**
Context/Coref	50.5% / 49.5%
Context/Base	**58.4% / 41.6%**

Table 6: Unilm comparison.

second comparison compares the human rephrasing, coreference-based, and context-based rephrasing. The results are presented in Table 5. The Human rephrasing outperforms all models, while the context achieves the closest performance, followed by the coreference and the base models. All results are statistically significant, except the context and human comparison. We used double evaluation of the human and coreference model rephrasing to calculate the judge's agreement, which resulted in a Kappa value of 0.604. Both the coreference and context model achieve close to human performance when the context-based model is almost equal to the human rephrasing.

6.2.2 Language Model Comperison

We choose BART as the base language model (LM) for our work because its denoising pre-train objective function seems suitable to deal with the noisy transcripts data. To evaluate this, we compare our results to another model - unilm (Bao et al., 2020). This model was trained as both masked LM and autoregressive LM and showed improvement on a variety of downstream tasks. To compare BART and unilm, we first compare our three candidate models using each of the LM, and then compare the best models with each LM. The results of the unilm variations are presented in Table 6. The hierarchy between the model variations remains, while the differences between models are smaller and become statistically insignificant. Nevertheless, we chose the context model as the best model for the cross LM comparison. The results are presented in Table 7. BART outperforms Unilm both on direct comparison and in its performance vs. the human rephrasing.

Models	Result
human/BART	50.8% / 49.2%
human/Unilm	53.1% / 46.9%
BART/Unilm	53.0% / 47.0%

Table 7: Cross LM comparison. Both models used the context annotations.

Models	Result
Human/Coref	56.1% / 43.9%
Human/Base	**68.2% / 31.8%**
Base/Coref	**40.1% / 59.9%**
Context/Coref	46.1% / 53.9%
Human/Context	54.3% / 45.6%

Table 8: Result on the public dataset. Using the base, coreference, and context models. Bold results are significant.

6.2.3 Public Dataset Results

The ID dataset contains sensitive and personal data and cannot be released to the public. In this subsection, we ran the same comparison of base/coreference/context and human annotators using the AIR-public dataset. These results can also highlight the dependency of the algorithm performance on the dataset size. The results are presented in Table 8. The human rephrasing is still the best model compared to all the others, but surprisingly it does not have a bigger gap in performance compared to the full dataset. We explain this by the mix of the different datasets. Each of the datasets contains a different distribution of AIs. Removing the ID dataset allows the model to overfit to the remaining existing datasets. The context model suffered the most from the reduced size of the dataset and is outperformed by the coreference model. We attribute this to the fact that the model trains twice on the data set (first for context detection, then for rephrasing). This shows that the coreference hints are best utilized when the dataset size is small.

7 Conclusion

In this work, we present the problem of action items rephrasing in meeting transcripts. We introduce a new dataset for the task and establish a baseline attributed to the BART transformer model. We then present two novel ways to considerably improve the baseline. Particularly, by collecting context annotations, and despite a relatively low agreement between the annotators, we are able to considerably improve the rephrasing model's performance. We evaluate our work by automated metrics, as well as independent human evaluators.

References

Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*.

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Songhao Piao, Jianfeng Gao, Ming Zhou, and Hsiao Wuen Hon. 2020. UNILMv2: Pseudo-masked language models for unified language model pre-training.

Paul N. Bennett and Jaime Carbonell. 2005. Detecting action-items in e-mail. In *SIGIR 2005 - Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Paul N. Bennett and Jaime G. Carbonell. 2007. Combining probability-based rankers for action-item detection. In *NAACL HLT 2007 - Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics, Proceedings of the Main Conference*.

Wauter Bosma. 2003. Query-based Summarization using Rhetorical Structure Theory. In *Proc. of CLIN 2003*.

Avi Caciularu, Arman Cohan, Iz Beltagy, Matthew E. Peters, Arie Cattan, and Ido Dagan. 2021. Cross-Document Language Modeling.

Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, Guillaume Lathoud, Mike Lincoln, Agnes Lisowska, Lain McCowan, Wilfried Post, Dennis Reidsma, and Pierre Wellner. 2006. The AMI Meeting Corpus: A pre-announcement. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.

Arie Cattan, Alon Eirew, Gabriel Stanovsky, Mandar Joshi, and Ido Dagan. 2020. Streamlining cross-document coreference resolution: Evaluation and modeling.

Jacob Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*.

Sergey Golitsynskiy Douglas Oard, William Webber, David A. Kirsch. 2015. Avocado Research Email Collection.

Arash Einolghozati, Anchit Gupta, Keith Diedrick, and Sonal Gupta. 2020. Sound Natural: Content Rephrasing in Dialog Systems.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2019. AllenNLP: A Deep Semantic Natural Language Processing Platform.

Nikhil Garg, Benoit Favre, Korbinian Reidhammer, and Dilek Hakkani-Tür. 2009. ClusterRank: A graph based method for meeting summarization. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*.

Andrew F Hayes and Klaus Krippendorff. 2007. Answering the call for a standard reliability measure for coding data. *Communication methods and measures*, 1(1):77–89.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: the 90% solution. In *Human Language Technology Conference of the NAACL, Short Papers*.

Adam Janin, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke, and Chuck Wooters. 2003. The ICSI meeting corpus. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving Pre-training by Representing and Predicting Spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, pages 188–197. Association for Computational Linguistics (ACL).

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pretraining for Natural Language Generation, Translation, and Comprehension.

C Y Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Proceedings of the workshop on text summarization branches out (WAS 2004)*.

Chu Cheng Lin, Michael Gamon, Dongyeop Kang, and Patrick Pantel. 2018. Actionable email intent modeling with reparametrized rnNs †. In *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach.

LSC. LSC - Legal Services Corporation: America's Partner for Equal Justice.

Nitika Mathur, Tim Baldwin, and Trevor Cohn. 2020. Tangled up in bleu: Reevaluating the evaluation of automatic machine translation evaluation metrics. *arXiv preprint arXiv:2006.06264*.

Yehudit Meged, Avi Caciularu, Vered Shwartz, and Ido Dagan. 2020. Paraphrasing vs coreferring: Two sides of the same coin.

Sudipto Mukherjee, Subhabrata Mukherjee, Marcello Hasegawa, Ahmed Hassan Awadallah, and Ryen White. 2020. Smart to-do: Automatic generation of to-do items from emails. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8680–8689, Online. Association for Computational Linguistics.

Preksha Nema, Mitesh M. Khapra, Anirban Laha, and Balaraman Ravindran. 2017. Diversity driven attention model for query-based abstractive summarization. In *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*.

Tatsuro Oya, Yashar Mehdad, Giuseppe Carenini, and Raymond Ng. 2014. A template-based abstractive meeting summarization: Leveraging summary and source text relationships. In *INLG 2014 - Proceedings of the 8th International Natural Language Generation Conference, including - Proceedings of the INLG and SIGDIAL 2014 Joint Session*, pages 45–53. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. BLEU: a method for automatic evaluation of machine translation. *ACL*.

Nazreena Rahman and Bhogeswar Borah. 2016. A survey on existing extractive techniques for query-based text summarization. In *2015 International Symposium on Advanced Computing and Communication, ISACC 2015*, pages 98–102. Institute of Electrical and Electronics Engineers Inc.

Owen Rambow, Lokesh Shrestha, John Chen, and Chirsty Lauridsen. 2004. Summarizing email threads.

Dennis Reidsma and Jean Carletta. 2008. Reliability measurement without limits. *Computational Linguistics*, 34(3):319–326.

Horacio Saggion, Kalina Bontcheva, and Hamish Cunningham. 2003. Robust generic and query-based summarisation. page 235.

Simon Scerri, Gerhard Gossen, Brian Davis, and Siegfried Handschuh. 2010. Classifying action items for semantic email. In *Proceedings of the 7th International Conference on Language Resources and Evaluation, LREC 2010*.

Wee Meng Soon, Daniel Chung Yong Lim, and Hwee Tou Ng. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*.

Rhea Sukthanker, Soujanya Poria, Erik Cambria, and Ramkumar Thirunavukarasu. 2020. Anaphora and coreference resolution: A review. *Information Fusion*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 2017-Decem, pages 5999–6009.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing.

Liyan Xu and Jinho D. Choi. 2020. Revealing the myth of higher-order inference in coreference resolution.

Wenyuan Zeng, Wenjie Luo, Sanja Fidler, and Raquel Urtasun. 2016. Efficient Summarization with Read-Again and Copy Mechanism.

Xingxing Zhang, Furu Wei, and Ming Zhou. 2020. Hibert: Document level pre-training of hierarchical bidirectional transformers for document summarization. In *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*.

A Test and Validation Sets Construction

Before evaluating the full test set, we ran a small subset of tests (60 instances overall) of the test set to get initial results. We compared the human rephrasing to the base model.The results are surprising, where most base models scored higher than the human rephrasing. While these results are attractive, manual examination showed that the base model often made mistakes that a human annotator could handle easily. Additionally, we checked the agreement between the judges on the same instances using Cohen's kappa (Cohen, 1960). The agreement was unexpectedly low 0.39. We explain these two phenomena by offering the hypothesis - some samples are very hard / impossible to rephrase properly. We suggest three reasons:

Context not in the sample In these samples the required context is not found in the part of the transcript that is included in the sample (each sample includes seven sentences before the AI, and three after).

Context requires real-world knowledge In some samples, some real-world knowledge is required to understand the AI. For example, our internal dataset is composed of technical team meetings that require prior knowledge in programming, software engineering, and NLP. Our annotators occasionally lack the technical knowledge required to understand the full AI meaning.

Rephrasing is hard Some samples were very hard to rephrase, even for humans, and often require a number of passes over the text to rephrase properly. Even though all of our annotators are proficient in English, they still had a very hard time rephrasing part of the questions. This results in bad rephrasing.

Surprisingly, when both the model and human fail, the human judge tends to prefer the model output. We attribute this to the fact that the model output was always grammatically and semantically correct, while the human had a tendency to write malformed sentences when the rephrasing was unclear.

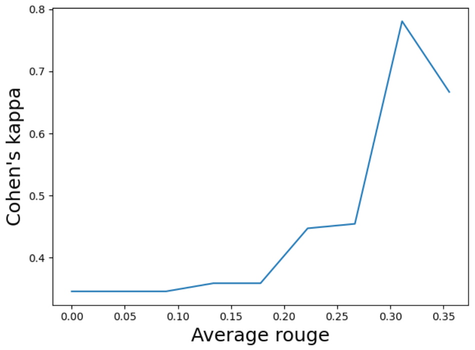


Figure 4: Cohen's Kappa as a function of a threshold serving as lower bound on the samples' average rouge.

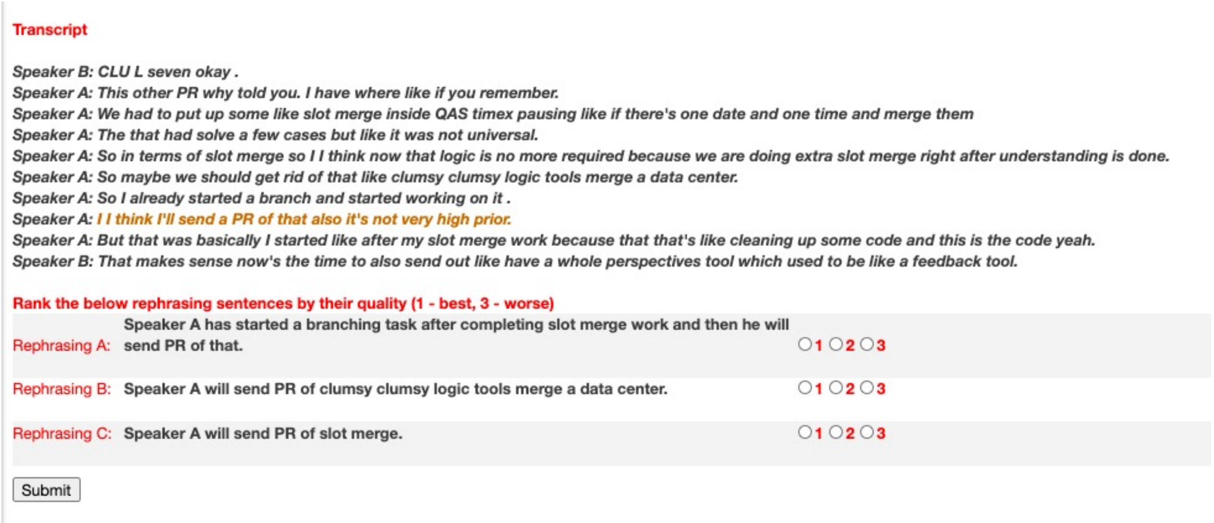
Rephraseable instances In cases where human annotators produce relatively similar rephrasing, we can presume the model will do the same, and thus we can think of those samples as 'easy to rephrase', or *rephraseable*. In order to find these, we take the *average rouge score* between all pairs of human rephrasing, as follows:

$$score(s) = \frac{1}{|R_s|^2} \sum_{\langle r,r' \rangle \in R_s} rouge_2(r, r') \,, \quad (1)$$

where s is a input sample, R_s is the set of all human rephrases of s, and $rouge_2(\cdot, \cdot)$ is the Rouge-2 measure. As *rephraseables*, we consider half of the samples – those with higher average rouge per (1). Figure 4 shows that, as one may expect, by considering the more rephraseable samples (i.e., putting a lower bound on the samples' average rouge score), the agreement between the judges on the related task of marking relevant context increases.

We use these insights to construct the test set and validation set by randomly sampling instances with an average rouge score $\geq$ 0.3. In Section 6, we show that using this test set, the judges' agreement is significantly higher (0.61) on Cohen's Kappa.

B Rephrasing Evaluation UI



Transcript

Speaker B: CLU L seven okay .
Speaker A: This other PR why told you. I have where like if you remember.
Speaker A: We had to put up some like slot merge inside QAS timex pausing like if there's one date and one time and merge them
Speaker A: The that had solve a few cases but like it was not universal.
Speaker A: So in terms of slot merge so I I think now that logic is no more required because we are doing extra slot merge right after understanding is done.
Speaker A: So maybe we should get rid of that like clumsy clumsy logic tools merge a data center.
Speaker A: So I already started a branch and started working on it .
Speaker A: I I think I'll send a PR of that also it's not very high prior.
Speaker A: But that was basically I started like after my slot merge work because that that's like cleaning up some code and this is the code yeah.
Speaker B: That makes sense now's the time to also send out like have a whole perspectives tool which used to be like a feedback tool.

Rank the below rephrasing sentences by their quality (1 - best, 3 - worse)

Rephrasing A:	Speaker A has started a branching task after completing slot merge work and then he will send PR of that.	○1 ○2 ○3
Rephrasing B:	Speaker A will send PR of clumsy clumsy logic tools merge a data center.	○1 ○2 ○3
Rephrasing C:	Speaker A will send PR of slot merge.	○1 ○2 ○3

Figure 5: The UI used by human judges to evaluate rephrasing. The displayed models are (top to bottom) human rephrasing, base model, and coreference model.