

What Do You Mean ‘Why?’: Resolving Sluices in Conversations

Victor Petrén Bach Hansen,^{1,2} Anders Søgaard^{1,3}

¹Department of Computer Science, University of Copenhagen, Denmark

²Topdanmark A/S, Denmark

³Google Research, Berlin

{victor.petren, soegaard}@di.ku.dk

Abstract

In conversation, we often ask one-word questions such as ‘Why?’ or ‘Who?’. Such questions are typically easy for humans to answer, but can be hard for computers, because their resolution requires retrieving both the right semantic frames and the right arguments from context. This paper introduces the novel ellipsis resolution task of resolving such one-word questions, referred to as *sluices* in linguistics. We present a crowd-sourced dataset containing annotations of sluices from over 4,000 dialogues collected from conversational QA datasets, as well as a series of strong baseline architectures.

1 Introduction

Stand-alone *wh*-word questions, such as *When?* in Figure 1, are easy for us to understand, but in order to interpret them we need to retrieve implicit information from context. Learning to do so is an instance of *sluicing*, an ellipsis phenomenon, defined by Ross (1969) as ‘the effect of deleting everything but the preposed constituent of an embedded question, under the condition that the remainder of the question is identical to some other part of the sentence, or a preceding sentence.’ In the context of conversations, one-word *wh*-word questions are particularly frequent (Anand and Hardt 2016; Rønning, Hardt, and Søgaard 2018), and because they are often hard to resolve, they seem to be a frequent source of error in conversational question answering (Choi et al. 2018; Reddy, Chen, and Manning 2018) and dialogue understanding (Vlachos and Clark 2014). We refer to this type of sluicing as *conversational sluicing*.

Unlike previous work where sluice resolution is treated as predicting the span of the antecedent (Anand and Hardt 2016; Rønning, Hardt, and Søgaard 2018), we frame conversational sluice resolution as a Natural Language Generation (NLG) task, in which we seek to automatically generate the full question, given a question-answer context and a one-word question. To this end, we provide a novel corpus of conversational sluice annotations and explore a series of strong baselines and their performance on this dataset.

Q_1 : Where was the bombing?
 San Diego’s Edward J. Schwartz Federal
 A_1 : Courthouse.
 Q_2 : When?

 R_1 : When [*was the bombing?*]
 R_2 : When [*was the bombing of San Diego’s
 Edward J. Schwartz Federal Courthouse?*]

Figure 1: Example of conversational sluicing. Q_1 and A_1 provides a context for the second question Q_2 which has multiple correct resolutions, denoted in brackets, such as R_1 and R_2 .

Contributions In this paper we introduce the task of resolving conversational sluicing, a pervasive and challenging ellipsis phenomenon. We crowd-source a new dataset containing over 4000 annotated sluices, gathered from existing conversational QA datasets. We conduct a series of baseline experiments on this task, using both encoder-decoder frameworks, as well as language modelling objectives, and show through human evaluation of the predicted resolutions that these baselines are quite strong and at times even rival the quality of human annotators.

2 Background

Sluicing Ellipsis is the linguistic phenomenon that describes the omission of one or more words from a phrase that can be retrieved from a previous context. Sluicing is a case of ellipsis where content is elided from a question, leaving behind only the *wh*-remnant. Anand and Hardt (2016) and Rønning, Hardt, and Søgaard (2018) consider two types of sluices, namely *embedded* sluices and *root* sluices, also sometimes referred to as *bare* sluices.

- (1) My neighbor said he would stop by, but I don’t know when [*he would stop by*].
- (2)
 - a. My neighbor is stopping by.
 - b. When [*is the stopping by*]?

In Example (1), we see an instance of embedded sluicing where the question is a part of a larger structure, and (2)

is an example of a root sluice where the *wh*-fronted ellipsis is an utterance in itself, i.e. in a root environment. Anand and Hardt (2016) note that sluicing in dialogue often differs from sluicing in single-authored text, with root sluices being more prevalent in dialogue. In dialogue, using sluices – and ellipsis in general – requires a level of mutual understanding. Colman, Eshghi, and Healey (2008) therefore use ellipsis in dialogue as a means of quantifying mutual understanding in conversations.

Fernández, Ginzburg, and Lappin (2007) focus on the task of classifying occurrences of single-word sluices in conversations and call these *bare* sluices. They categorize such sluices into distinct categories; (i) *direct*, which is the case where the sluice queries for additional information that was quantified, either explicitly or implicitly, in the previous utterance; (ii) *reprise*, where the speaker is unable to understand an aspect of the previous utterance, which the initial speaker assumed as presupposed; (iii) *clarification*, where the speaker uses the sluice to ask for clarification of the entire preceding utterance; (iv) *Wh-anaphor*, where the antecedent is a *wh*-phrase; and (v) *unclear*, the case where it is difficult to understand what the sluice conveys, usually because of a lack of proper context. Note that the direct, reprise and clarification sluices are relatively easier to resolve, since their answer can always be retrieved from the previous sentence. Our corpus therefore ignores the first three types of conversational sluices and focuses on (*bare* or *stand-alone*) *wh*-anaphors; in our annotation experiments below, we also allow annotators to skip *unclear* instances. Similarly, Baird, Hamza, and Hardt (2018) presented classification experiments learning to distinguish between different types of sluices in dialogue.

Conversational sluices usually depend on their question-answer context, and can span both the previous utterances, i.e. the answer, as well as the previous question, whereas *direct/reprise/clarification* sluices only require retrieval of context from the previous utterance. Consider the multi-turn example:

- A: Did Ned have family?
B: Yes.
A: Who [*was Ned's family*]?

Resolving this sluice, depends on both the question initially asked by speaker A in addition to the outcome of the answer from speaker B. Looking only at the previous utterance, in this case, would not provide sufficient context, as the *Yes/No* utterance of speaker B determines what information from speaker A is relevant for the resolution.

The first efforts to resolve (non-conversational, standard) sluices, by identifying the antecedent of the *wh*-remnant, is due to Anand and McCloskey (2015), who describe a linguistically-informed annotation scheme for resolving sluices. They present a dataset of 3,100 annotated examples of sluices extracted from the New York Times section of the English Gigaword corpus. Anand and Hardt (2016) presented the first sluice resolution system, achieving decent performance, but Rønning, Hardt, and Sjøgaard (2018) subsequently presented a neural multi-task architecture outperforming their original model by some margin.

A few researchers have explored ellipsis resolution in dialogue: Kazuhide and Eiichiro (1998) discussed the importance of being able to resolve sluices to understand dialogue. They showed that for certain types of conversational ellipsis, it is possible to achieve good results with simple classification algorithms. Their results are not comparable to other results in the literature, because they focus on a small subset of phenomena, rely on linguistic preprocessing, and consider ellipsis phenomena in Japanese. Rønning, Hardt, and Sjøgaard (2018) also evaluate on conversational data from English Open Subtitles. Their results suggest that resolving sluices in dialogue is harder than domains such as newswire, with F_1 resolution scores dropping from > 0.7 in newswire to around 0.5 for conversations. As stated, these previous approaches to sluice resolution differs from ours, as we seek to generate a reconstruction of the sluice, not predict the span of the antecedent. Due to the fact that in a conversational context, the antecedent is conditioned on the response to the initial question in our question-answer context, it often results in disjoint antecedent spans, which cannot be represented in the architecture proposed by Rønning, Hardt, and Sjøgaard (2018). The advantage of resolving the sluice using NLG approaches is that for most downstream purposes, a fluent paraphrase of the *wh*-word and the antecedents is preferred and not only an antecedent span, that as stated above, can be non-coherent.

Question Generation Researchers have worked on question generation from text paragraphs (Zhao et al. 2018), relative clauses (Khullar et al. 2018), SQL queries (Guo et al. 2018), knowledge bases (Serban et al. 2016), etc. Khullar et al. (2018), which is probably the problem set-up most similar to ours, albeit much simpler, consider relative clauses such as in *I am giving fur balls to John who likes cats*. Their simple observation is that relative clauses translate almost straight-forwardly into questions, e.g., *Who likes cats?*. Using a small set of heuristic rules, they extract relative clauses and use them to generate training data for machine comprehension. Our task is considerably harder, since we deal with an ellipsis phenomenon that requires us to find antecedents in the previous dialogue turns. Our approach is also very different. While Khullar et al. (2018) can solve their problem with simple rules, we cannot, and we therefore present neural baseline architectures originally developed for language modeling and transduction tasks.

3 A Conversational Sluicing Dataset

In this work, we present a crowd-sourced annotated sluicing dataset. The dataset consists of sluice occurrences in conversational question answering contexts. The conversations are teacher-student dialogues, where the teacher asks questions about a background text passage, and the student has to answer the teacher's questions. Sluices, and ellipsis in general, are frequent in the data. Each datapoint consists of (i) an initial question, Q_1 , (ii) an answer to Q_1 , A_1 , together forming the QA context (Q_1, A_1) , (iii) a one word follow-up *wh*-question, Q_2 , (iv) a gold annotated resolution, R , to the sluice in (iii), written in free-text. The resolutions are what

we crowd-source to construct the new conversational sluicing dataset. Given question-answer context pairs (Q_1, A_1) and one-word follow-up questions Q_2 , we seek to resolve conversational sluices by generating the full questions R by explicitly generating the elided context, therefore framing it like a NLG task, rather than an antecedent selection task as done by Rønning, Hardt, and Søggaard (2018) and Anand and Hardt (2016). This also dramatically simplifies the annotation process as we only seek a resolved sluice in the form of R instead of the annotation scheme used by Anand and McCloskey (2015) and Rønning, Hardt, and Søggaard (2018), i.e. explicitly annotating the antecedent, sluiced expression, main predicate of the antecedent clause as well as potential correlates in addition to annotations for the auxiliary tasks.

This section describes the process of collecting and cleaning the annotations, and presents a quantitative and qualitative analysis of the dataset.

Data Collection Methodology In order to obtain our conversational sluicing dataset, we crawl existing conversational QA datasets, namely QuAC¹ and CoQA,² for question-answer contexts with one-word follow-up questions. Specifically, we identify all occurrences of five one-word questions: *Why?*, *What?*, *Where?*, *Who?* and *When?*. For each such question, we construct a tuple of the previous QA context and the follow-up question. This process results in roughly 4200 examples of conversational sluices.

We then proceeded to ask Amazon Mechanical Turkers (AMT) to fill out the remainder of the question as asked by the interrogator based on the the question-answer context pair. In order to not impose too many restrictions on the annotators, we left it up to the AMT workers to decide how much of the elided information they wanted to include in their answer, as a conversational sluice can often be solved in multiple ways. For example, in Figure 1 we consider both R_1 and R_2 as correct resolutions to the conversational sluice, even if R_1 did not specify the PPN *San Diego's Edward J. Schwartz Federal Courthouse* as the location of the bombing. In general, annotations often differed in whether modifiers and relative clauses were included, whether or not previous anaphora was resolved, etc. If the previous question and answer did not provide enough context to fill out the elided information, the workers were informed to simply skip it and move on to the next example. We collected a single annotation for each sluice in the training and test splits, and three annotations for each sluice in the test set. For the test set, we use each unique annotation as a separate datapoint. We allocated 1 minute per annotation and paid the workers \$0.13 for each accepted annotation. The average time spent per assignment was around 20 seconds, which results in an hourly rate of \$23.4. The total cost of the crowdsourcing process was \$797.

For our final corpus, we filter out the examples skipped by the annotators, in addition to the conversational sluices whose Q_1 context is less than 3 words, as these showed empirically to not contain enough information, usually due to

Q_1 being a sluice itself. Consider, for example:

Q_1 : By who?
 A_1 : Unknown assailants.
 Q_2 : Where?

Without first resolving the sluice *By who?*, we are unable to properly identify the antecedent, as it is unclear whether or not Q_2 refers to the current location of the assailant or the location of the actual assault. These are also the sluices categorized as *Unclear* by Fernández, Ginzburg, and Lappin (2007). After cleaning, we reduced the initial size from 4980 to 4175 datapoints.

Corpus Statistics In Table 1, we show the distribution of the different *wh*-questions across the various splits in our corpus. The dataset contains both instances of conversational sluices as well as reprise/direct/clarification sluices. We release the raw annotated version of the conversational sluicing corpus, as well as our cleaned version which we report our results on, including the splits used.³

Split	Why	Where	Who	What	When	Total
train	851	714	513	302	702	3082
val	84	71	54	39	52	300
test	229	183	97	83	201	793
Total	1164	968	664	424	955	4175

Table 1: Statistics of the *wh*-word distribution across the different splits for our conversational sluicing dataset.

Empirically, we did not observe many long distance dependencies between the sluice and corresponding antecedent, as it was found within a three-turn window a majority of the time (around 95%). Rønning, Hardt, and Søggaard (2018) similarly reports that long term dependencies (3 or more sentences between sluice and antecedent) are very rare (around 1%). Solving these rare dependencies would also be an interesting task, but is however outside the scope of this work. This dataset provides a reasonable limitation for a stab at an already challenging phenomenon.

Performance Metrics Natural language generation systems are often evaluated in terms of BLEU scores (Papineni et al. 2002) and on subsamples of standard corpora. Neither are likely to be optimal. Finding an appropriate performance metric that correlates with human judgments of resolution quality, is crucial to ensure progress on conversational sluicing resolution; and evaluating across different samples is equally important to avoid community-wide over-fitting to one particular sample. We hope to be able to contribute to improving both performance metrics and the data situation, but for now we also report the performance of our baseline systems in terms of BLEU scores on a random subsample. In order to combat the bias introduced by BLEU, we supplement the scores with alternative performance metrics, as well as with human judgments from professional

¹<https://quac.ai/>

²<https://stanfordnlp.github.io/coqa/>

³https://github.com/vpetren/conv_sluice_resolution

Model	GLEU	BLEU	CHRF	CHRP	CHRR
C&E Q1	0.035	0.043	0.114	0.034	0.166
C&E A	0.010	0.016	0.034	0.011	0.048
LSTM-SEQ2SEQ	0.232	0.304	0.276	0.311	0.274
TRANSFORMER	0.337	0.391	0.443	0.461	0.442
GPT-2	0.067	0.117	0.138	0.109	0.167
GPT-2 (FT)	0.348	0.391	0.467	0.499	0.470
ANN AGREE	0.570	0.589	0.712	0.704	0.720

Table 2: Results on our conversational sluicing dataset for a series of baseline architectures. We measure the performance using BLEU, GLEU and character n -gram F-score, precision and recall on the test split. In the last row, ANN AGREE denotes the inter-annotator agreement as the average between two randomly sampled gold annotations from each data point of the test set.

annotators. BLEU originally was intended for corpus-level evaluation and has several limitations when applied at the sentence-level (Rapp 2009). We therefore also include the GLEU metric, as proposed by (Wu et al. 2016), which according to their experiments, is better suited for sentence-level evaluation, while still correlating well with BLEU on the corpus-level.⁴ In addition to BLEU and GLEU we also measure the the character n -gram F-score (CHRF) (Popović 2015), as well as the precision (CHRP) and recall (CHRR). We use $\beta = 3$, i.e. assigning a higher weight to recall, as it has been shown to correlate better with human judgements than other popular automatic machine translation metrics, such as BLEU and ROGUE-L. For n we use 4-grams.

Given the shortcomings of automatic evaluation metrics, we also include a human evaluation study. We sample n contexts along with the gold sluice resolution and the resolutions generated by our baseline models from the test set and ask human evaluators to rank them according to relative quality. We obtained judgments of 100 document instances and report on these experiments in §5.

Annotation Quality In the last row of Table 2, ANN AGREE, we report the inter-annotator agreement scores of the test set. For each of the 3 collected annotation per conversational sluice instance, we sample 2 of them and calculate BLEU, GLEU, chrF, chrP and chrR scores between them as a measurement of annotator agreement. As different annotations can be considered correct sluice resolutions, we use this measurement as a means to set an expectation for the performance ceiling of our models. In general we observe that there seems to be reasonably high annotator agreement scores compared to the best performing models, but still indicates that the sluices can be solved in multiple correct ways.

4 Experiments

In our experiments, we use the splits outlined in Table 1 (also made publicly available). We preprocess our data by append-

⁴We use the sentence-level GLEU and BLEU implementations provided by NLTK with the smoothing function introduced by Lin and Och (2004)

Model	MRR	r_1
LSTM-SEQ2SEQ	0.295	0.005
TRANSFORMER	0.381	0.030
GPT-2 (FT)	0.529	0.190
GOLD	0.879	0.775

Table 3: The results of the human judgement experiment. To obtain human judgments, we asked three annotators to rank the output of three systems and the crowd-sourced gold annotations. MRR is the mean reciprocal ranking, and r_1 refers to the fraction of presented examples where the model was ranked as number 1. Our results show that the fine-tuned GPT-2 model produces favorable resolutions, both in terms of automatic as well as human evaluation and 1/5 instances *better* than gold annotations.

ing the QA context and one-word question together, converting the input sequence into the format $\langle s \rangle Q_1 \langle del \rangle A_1 \langle del \rangle Q_2 \langle /s \rangle$ and the target sequence we seek to generate as $\langle s \rangle R \langle /s \rangle$. Here $\langle del \rangle$ is a special delimiter token, and $\langle s \rangle$ and $\langle /s \rangle$, denote the beginning and end of the sequence. In addition to this, we only preprocess the data by performing lower-casing and tokenization.

4.1 Baseline models

In this section, we present a number of different baseline architectures and heuristics for the task of conversational sluice resolution.

Copy & Edit Heuristics Seeing as the structure of the resolved sluice in some cases takes on the form of either Q_1 , especially in the cases where a yes/no answer precedes it, or A , as seen in Figure 1, we propose two simple copy and edit heuristics. (i) Given the QA-context and our conversational sluice Q_2 , we simply replace the wh-question word in Q_1 with Q_2 and use this augmented question as the resolution to our sluice. We refer to this as C&E Q1. (ii) Similarly, we can copy the answer from A and prepend the Q_2 sluice to it. We refer to this as C&E A.

LSTM-seq2seq Sequence-to-sequence models (Sutskever, Vinyals, and Le 2014), or seq2seq, have previously been successfully applied to conversational modelling tasks (Vinyals and Le 2015). They use the encoder-decoder framework, where an input context is encoded by an encoder-module, usually a variant of Recurrent Neural Networks (RNNs), and decoded by a decoder-module, into the target sequence. For both the encoder and decoder, we use a standard two-layer LSTM (Hochreiter and Schmidhuber 1997), with a hidden state size of 512, and regularized using a dropout rate of 0.5. We initialize the embedding matrix with 300 dimensional GloVe (Pennington, Socher, and Manning 2014), which remains fixed during training. We optimize the end-to-end network using Adam (Kingma and Ba 2014), with the

default learning rate of 0.001.⁵

Transformer The transformer architecture (Vaswani et al. 2017) is now the *de facto* standard architecture in machine translation and has paved the way for state-of-the-art pre-trained contextual language encoders such as BERT (Devlin et al. 2018) and the OpenAI GPT-2 (Radford et al. 2019). While still adopting the encoder-decoder framework, instead of processing the source and target sequences sequentially, it relies on a multi-headed self-attention mechanism, attending over the entire sequence at same time, allowing for greater parallelization and a positional encoding of the sequence, ensures that contextual information is maintained. As our conversational sluicing resolution corpus is small in comparison to the corpora used in the experiments by Vaswani et al. (2017), we limit ourselves to three encoder/decoder layers to 3 (compared to 6 in their work), after observing improvements on our validation data.⁶ As with the LSTM-seq2seq model, we initialize the embedding matrix with 300 dimensional GloVe embeddings, but otherwise we use the default hyperparameters.

GPT-2 The Generative Pretrained Transformer-2 (GPT-2) (Radford et al. 2019), trained to simply predict the next word in 40GB of Internet text, has since its introduction been used to generate state-of-the-art performance on multiple language modelling datasets. The GPT-2 architecture, as mentioned above, is based on the transformer architecture. In our experiments, we use the small pretrained model released by OpenAI (117M parameters). We experiment both with the pretrained GPT-2 model as is, as well as with fine-tuning it on our sluicing corpus. When fine-tuning the model, we simply concatenate the input and output sequences together and input them to the language model. Unlike the LSTM-seq2seq and Transformer, we do not fine-tune the GPT-2 model until convergence, but instead we ran it for 18 hours on an Nvidia TitanX GPU. We also report the performance of the GPT-2 model on our task when no fine-tuning has taken place.

Other baselines considered Inspired by Hill, Cho, and Korhonen (2016) and Lample, Denoyer, and Ranzato (2017), we also experimented with pretraining the SEQ2SEQ-LSTM and TRANSFORMER architectures with sequential de-noising autoencoder objectives. We collected a dataset consisting of 350.000 questions from CoQA, QuAC and SQuAD 2.0, making sure not to include cases of sluices, hypothesizing that this would allow the encoder and decoder to learn the internal structure and representation of questions. After pre-training, we fine-tune the architectures on our conversational sluicing data. These experiments did, however, not lead to any improvements in the performance when using automatic metrics. A manual inspection of the

⁵Implementation is based on <https://github.com/bentrevett/pytorch-seq2seq>.

⁶Implementation is based on <https://github.com/jadore801120/attention-is-all-you-need-pytorch/>

generated resolutions did not reveal any noticeable improvements over their non pre-trained counterparts, so we do not report the results below.

Again, we stress that due to the reasons listed above, i.e. incompatible annotation schemes between our work and that of Rønning, Hardt, and Sjøgaard (2018) as well as the lack of flexibility that a span-prediction model provides, we do not use their work as a baseline. We hypothesize that our heuristics, C&E Q1 and C&E A, will serve as an indication as to what we can expect from these types of models.

4.2 Results

Table 2 summarizes the results from our baseline models on our conversational sluicing corpus, using standard automatic performance metrics. The results suggest that the fine-tuned GPT-2 architecture is superior to all other baselines across the board, achieving scores closest to the inter-annotator ceiling, with the TRANSFORMER model rivalling it on the BLEU score. Although the C&E Q1 and C&E A heuristics could seem like strong baselines, as some of the examples in Table 4 and Figure 1 might suggest, our results tell a different story. Again, this illustrates the flexibility that is required to resolve these conversational sluices, which a non-disjoint antecedent span fails to capture. We can observe that without the task-specific fine-tuning, the GPT-2 model falls short, as it ultimately just proceeds to generate what comes after the sluice, not resolving it. However, this extensive pretraining does shine through compared to the TRANSFORMER model, when fine-tuned on our dataset as we also can see from our human evaluation (illustrated in 3), which we discuss in the next section.

5 Analysis

Human judgment of generated resolutions Knowing that our automatic evaluation metrics can be biased when applied at the sentence-level, we also include a human evaluation study on a random sample of 100 instances of sluices. We asked human evaluators to rank the resolutions generated by our best performing models, i.e. the LSTM-SEQ2SEQ architecture, the TRANSFORMER architecture, our fine-tuned GPT-2 model, as well as the human annotators' resolutions, by their quality and relevance in a QA context. We presented the four resolutions in random order and asked subjects to place them, from best to worst. If they deemed two or more candidates to be equally good or bad, we instructed them to simply order these randomly. We report performance using the Mean Reciprocal Rank (MRR), and what we refer to as r_1 , which denotes the fraction of presented examples where the model was ranked as number 1. Our evaluation, shown in Table 3, reveals that the human judges tend to favour the resolutions provided by GPT-2 (FT) over the ones produced by the TRANSFORMER architecture. In fact, the GPT-2 resolutions are chosen over all other resolutions, including our gold standard, in 1/5 instances. Generally we see the same trend in the human evaluation experiment as with the automatic metrics, except that the GPT-2 model now significantly outperforms the other baselines. We believe this can be attributed to the fact that our human judges may be bi-

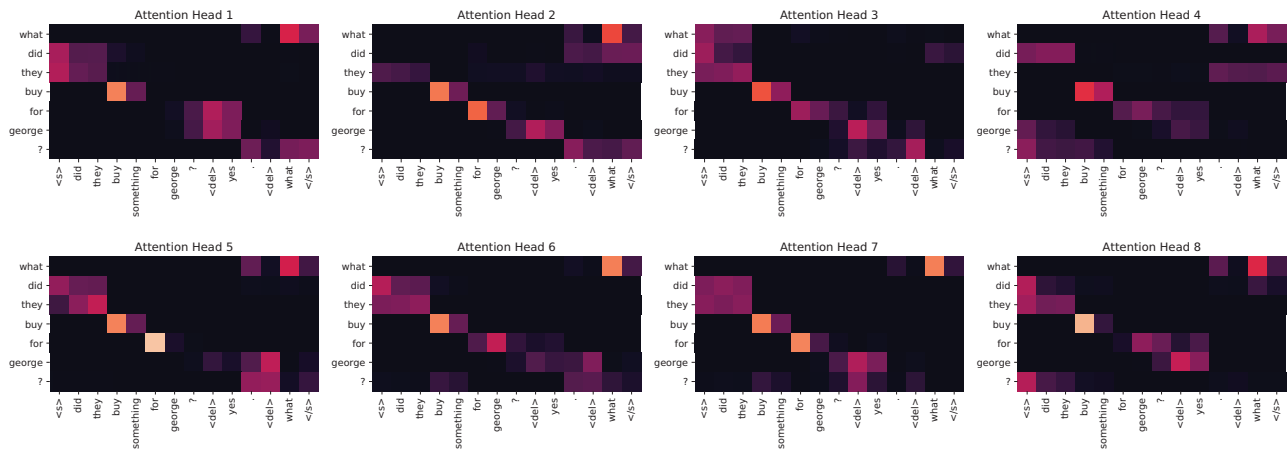


Figure 2: Illustration of the attention weights from all the 8 attention heads in the final decoder layer of the Transformer network. The x-axis corresponds to the position in the input sequence, whereas the y-axis corresponds to the output sequence.

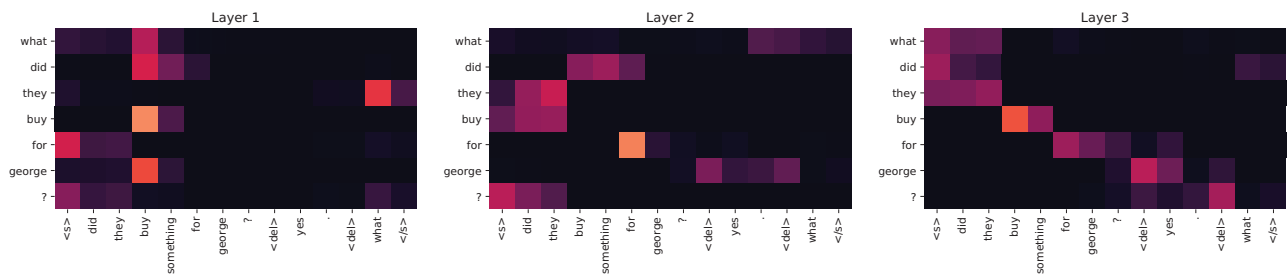


Figure 3: Illustration of the attention weights from a single attention head in the 3-layer Transformer network, during decoding. The x-axis corresponds to the position in the input sequence, whereas the y-axis corresponds to the output sequence.

ased toward selecting well-formed resolutions, and the GPT-2 language model may simply be better at generating fluent language.

To illustrate an instance where GPT-2 can generate a more expressive resolution than our gold standard, consider the example in Figure 4. Here, the fine-tuned OpenAI GPT-2 model generates a resolution that the judges found to be better than the gold standard, not because the gold-standard was wrong, but because the automatic resolution was more informative, easing interpretation.

Q_1 : Is anyone who works with them mentioned?
 A_1 : Yes.
 Q_2 : Who?

 R_{GPT2} : Who [else is mentioned]?
 R_{Gold} : Who [is mentioned]?

Figure 4: Conversational sluice resolution by the fine-tuned GPT-2 model that is judged better than the gold standard by our annotators.

Visualization of attention weights An advantage of the attention mechanism, is that it allows for high interpretability, when it comes to the showing where in the input se-

quence the model is attending at a given time-step. To get a better understanding of where the Transformer attends during decoding, we visualize the internal attention mechanisms of the model trained on our conversational sluicing corpus. Figure 3 shows the attention matrix heatmaps of a single attention-head in each layer and Figure 2 shows the attention matrix heatmaps for each of the 8 attention-heads in the last layer of the Transformer. When looking at Figure 3, we see that the various layers encode different levels of information, with the attention-head of the last layer seemingly being the most structured. From Figure 2, we can observe that the various attention-heads mostly present the same pattern. When generating the first word of the resolution, the attention is at the end of the input sequence, i.e. on the *wh*-fronted ellipsis. Generating the subsequent tokens then shifts the attention back to the beginning of the input sequence and learns to integrate the information of the question-answer context, as the resolution of the conversational sluice tends to repeat the structure of the antecedent of both the question and answer.

Inspection of model output Table 4 present examples of conversational sluices from the test set along with the resolutions generated by our baselines as well as a gold annotated resolution. From the examples, we can observe that

CONTEXT	LSTM-S2S	TRANSFORMER	GPT-2	GOLD
Q_1 : What did Susie do? A_1 : Woke up. Q_2 : When?	When did they go??	When did Susie woke up?	When did Susie wake up?	When did Susie wake up?
Q_1 : Did the island ever change its form of government? A_1 : Yes. Q_2 : When?	When did the objective of the?? Scotland?	When did the island change its form of government?	When did the island form?	When did the island change its form of government?
Q_1 : Is there any mysterious character? A_1 : Yes. Q_2 : Who?	Who is the other??	Who are the character in?	Who was the famous person that was added to the story?	Who is the mysterious character?
Q_1 : Did he say anything before leaving? A_1 : Yes. Q_2 : What?	What did he do ?	What did he say?	What did he say before he left?	What did he say?

Table 4: Generated output from our series of baselines, given a question-answer context, (Q_1, A_1) and follow-up one-word question. Examples are taken from the test split.

the LSTM-SEQ2SEQ often produces more nonsensical and less grammatically correct sentences, e.g. overusing question marks and inserting them in the middle of the sentences and it generally performs best when the input context and resolutions are short. The output of the TRANSFORMER does improve upon the results of the LSTM-SEQ2SEQ, producing more correct and coherent sentences, however, the lack of pretraining compared to GPT-2, still results in less expressive sentences. Most impressive are the results from the fine-tuned GPT-2 model. From its r_1 value we can see that almost 20% of the instances, it actually generates a sluice resolution that our human judges ranked higher than the gold resolution. E.g., in the last sample generated by the GPT-2 model, demonstrates how it is able to incorporate all the information of the initial question Q_1 , to a much higher degree than the what the annotator noted. The extensive pretraining does however allow the generated output to deviate a bit too much from the objective, as seen in the 3rd row.

Applying sluice resolutions in QA systems As mentioned in §1, the ability to resolve occurrences of ellipsis, either implicitly or explicitly, is important for question-answering system. With our gold annotated sluice resolutions, we replace instances of conversational sluices in the CoQA development set with their resolved counterparts, and evaluate the quality of the answers their baseline model provides.⁷ In Figure 5, we see how the resolution of the conversational sluice leads to a much better answer, $A_{no-sluice}$, compared to the case where the model has to automatically draw the connection between ‘Why?’ and the context in Q_1 and A_1 . Of course injecting our annotations into the input at test time also biases the input data, making it less similar to the training data, and for this reason resolving sluices this

⁷Code for the pretrained CoQA baseline model is provided by <https://github.com/stanfordnlp/coqa-baselines>

Q_1 : What did Valetta think Mysie mustn’t do?
 A_1 : Stay out after dark.
 Q_2 : Why [*does Valetta think that Mysie shouldn’t stay out after dark*]?

 $A_{no-sluice}$: For fear she should cough.
 A_{sluice} : no.
 A_{gold} : Fear she should cough.

Figure 5: A case where resolving the sluice in the an instance of the CoQA dataset improves the performance of QA system. $A_{no-sluice}$ is the answer generated when information contained in the bracket is included.

way did not lead to significant improvements on average.

6 Conclusion

This paper addresses the challenge of resolving occurrences of conversational sluices; that is, correctly identifying the antecedent of a bare *wh*-fronted ellipsis in a dialogue setting. We frame the task as a language generation task, where we seek to generate the elided material. To this end, we crowd-sourced a new dataset of conversational sluices. We evaluate the performance of encoder-decoder architectures and language models on this data and show that human judges favour the resolutions generated by GPT-2, fine-tuned on our crowd-sourced annotations. Interestingly, resolutions rival the quality of human annotations.

References

- Anand, P., and Hardt, D. 2016. Antecedent selection for sluicing: Structure and content. 1234–1243.
- Anand, P., and McCloskey, J. 2015. Annotating the implicit content of sluices. In *LAW@NAACL-HLT*.

- Baird, A.; Hamza, A.; and Hardt, D. 2018. Classifying sluice occurrences in dialogue. In *Proceedings of the 11th Language Resources and Evaluation Conference*. Miyazaki, Japan: European Language Resource Association.
- Choi, E.; He, H.; Iyyer, M.; Yatskar, M.; Yih, W.; Choi, Y.; Liang, P.; and Zettlemoyer, L. 2018. Quac : Question answering in context. *CoRR* abs/1808.07036.
- Colman, M.; Eshghi, A.; and Healey, P. 2008. Quantifying ellipsis in dialogue: an index of mutual understanding. In *Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue*, 96–99. Columbus, Ohio: Association for Computational Linguistics.
- Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR* abs/1810.04805.
- Fernández, R.; Ginzburg, J.; and Lappin, S. 2007. Classifying non-sentential utterances in dialogue: A machine learning approach. *American Journal of Computational Linguistics* 33(3):397–427.
- Guo, D.; Sun, Y.; Tang, D.; Duan, N.; Yin, J.; Chi, H.; Cao, J.; Chen, P.; and Zhou, M. 2018. Question generation from sql queries improves neural semantic parsing. In *EMNLP*.
- Hill, F.; Cho, K.; and Korhonen, A. 2016. Learning distributed representations of sentences from unlabelled data. *CoRR* abs/1602.03483.
- Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Comput.* 9(8):1735–1780.
- Kazuhide, Y., and Eiichiro, S. 1998. Feasibility study for ellipsis resolution in dialogues by machine-learning technique. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 2, ACL '98/COLING '98*, 1428–1435. Stroudsburg, PA, USA: Association for Computational Linguistics.
- Khullar, P.; Rachna, K.; Hase, M.; and Shrivastava, M. 2018. Automatic question generation using relative pronouns and adverbs. In *ACL*.
- Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *ICLR*.
- Lample, G.; Denoyer, L.; and Ranzato, M. 2017. Unsupervised machine translation using monolingual corpora only. *CoRR* abs/1711.00043.
- Lin, C.-Y., and Och, F. J. 2004. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics, ACL '04*. Stroudsburg, PA, USA: Association for Computational Linguistics.
- Papineni, K.; Roukus, S.; Ward, T.; and Zhu, W.-J. . 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 311–318.
- Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.
- Popović, M. 2015. chrF: character n-gram f-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, 392–395. Lisbon, Portugal: Association for Computational Linguistics.
- Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language models are unsupervised multitask learners.
- Rapp, R. 2009. The back-translation score: Automatic mt evaluation at the sentence level without reference translations. In *ACL*.
- Reddy, S.; Chen, D.; and Manning, C. D. 2018. Coqa: A conversational question answering challenge. *CoRR* abs/1808.07042.
- Rønning, O.; Hardt, D.; and Sjøgaard, A. 2018. Sluice resolution without hand-crafted features over brittle syntax trees. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 236–241. New Orleans, Louisiana: Association for Computational Linguistics.
- Ross, J. R. 1969. Guess who? *CLS 5: Papers from the Fifth Regional Meeting of the Chicago Linguistic Society*.
- Serban, I. V.; Garcia-Duran, A.; Gulcehre, C.; Ahn, S.; Chandar, S.; Courville, A.; and Bengio, Y. 2016. Generating factoid questions with recurrent neural networks: The 30m factoid question-answer corpus. In *EMNLP*.
- Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. *CoRR* abs/1409.3215.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. *CoRR* abs/1706.03762.
- Vinyals, O., and Le, Q. V. 2015. A neural conversational model. *CoRR* abs/1506.05869.
- Vlachos, A., and Clark, S. 2014. A new corpus and imitation learning framework for context-dependent semantic parsing. In *TACL*.
- Wu, Y.; Schuster, M.; Chen, Z.; Le, Q. V.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; Klingner, J.; Shah, A.; Johnson, M.; Liu, X.; Kaiser, L.; Gouws, S.; Kato, Y.; Kudo, T.; Kazawa, H.; Stevens, K.; Kurian, G.; Patil, N.; Wang, W.; Young, C.; Smith, J.; Riesa, J.; Rudnick, A.; Vinyals, O.; Corrado, G.; Hughes, M.; and Dean, J. 2016. Google’s neural machine translation system: Bridging the gap between human and machine translation. *CoRR* abs/1609.08144.
- Zhao, Y.; Ni, X.; Ding, Y.; and Ke, Q. 2018. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *EMNLP*.