Resolving Implicit References in Instructional Texts

Talita Rani Anthonio **Michael Roth**

University of Stuttgart
Institute for Natural Language Processing
{anthonta,rothml}@ims.uni-stuttgart.de

Abstract

The usage of (co-)referring expressions in discourse contributes to the coherence of a text. However, text comprehension can be difficult when referring expressions are non-verbalized and have to be resolved in the discourse context. In this paper, we propose a novel dataset of such *implicit references*, which we automatically derive from insertions of references in collaboratively edited how-to guides. Our dataset consists of 6,014 instances, making it one of the largest datasets of implicit references and a useful starting point to investigate misunderstandings caused by underspecified language. We test different methods for resolving implicit references in our dataset based on the Generative Pre-trained Transformer model (GPT) and compare them to heuristic baselines. Our experiments indicate that GPT can accurately resolve the majority of implicit references in our data. Finally, we investigate remaining errors and examine human preferences regarding different resolutions of an implicit reference given the discourse context.

1 Introduction

Implicit language phenomena can be challenging for both human and machine processing. For example, references play a crucial role in instructional texts as they provide answers to questions such as *Which objects need to be used?* If such references are not made explicitly, they might be clear to readers who have task-specific knowledge, but for others they might cause problems or misunderstandings. Resolving such *implicit references* could improve clarity and prevent problems in discourse processing when multiple interpretations exist.

In natural language processing, implicit references have been handled as part of existing tasks such as semantic role labeling of implicit arguments (Gerber and Chai, 2012, cf. §3). Implicit arguments are generally hard to model computationally because they do not show up in easy to learn surface patterns (Ruppenhofer et al., 2009). The use of role-semantic formalisms further complicates progress in this direction because manual annotation requires trained annotators and previous training datasets have been comparatively small. For example, most datasets of implicit arguments consist of just hundreds of instances or only predicate-specific annotations (Moor et al., 2013).

Defrost Ground Beef
(1) Place the ground beef in a microwave.
(2) Microwave until it finishes thawing.
(3) Use _____ within 1 or 2 days.
(3') Use the beef within 1 or 2 days.
(3") Use the microwave within 1 or 2 days.

Table 1: Simplified example based on the wikiHow article "Defrost Ground Beef": sentences (1–3) show the original version of a text. Sentence (3') and (3") show revised versions that include a manually inserted or automatically generated reference (see §5), respectively.

We propose a task and dataset of implicit references which we obtain without manual annotation. Specifically, we create a dataset by extracting insertions of references in the revision history of collaboratively edited how-to guides. Previous work has shown that revisions in instructional texts are typically made to improve a text (Anthonio and Roth, 2020). Based on this observation, we assume that explicit references are inserted when an implicit reference is perceived as problematic in discourse context. A simplified example from our dataset and an illustration of our task are provided in Table 1.

As shown by the example, our data consists of insertions of single references and the task is to predict these inserted references. As a benefit over existing work, the task does not depend on any formalism of role semantics, which means that models can be evaluated in an end-to-end setting.

As a dataset for the proposed task, we provide

6,014 instances of implicit references, which we extracted automatically by comparing different versions of articles in wikiHow[1]. In practice, we make use of an existing resource of wikiHow sentences and revisions called wikiHowToImprove (Anthonio et al., 2020), from which we select specifically those cases in which a *referring expression* was inserted that refers to an *entity* mentioned in the preceding context. Based on this dataset, we set up a cloze task in which we evaluate the ability of computational models to generate references for insertions that occur naturally in publicly available texts. Finally, we analyze predictions of different modeling approaches as well as differences between model-generated and human-inserted references, which provide useful insights regarding potential weaknesses of existing models and potential causes of human misunderstandings.

In sum, we make the following contributions:

- We propose a new task that requires NLP models to generate explicit references to resolve cases of implicit language (§2).
- We provide a dataset of 6,014 texts that involve the insertion of an explicit reference according to the text's revision history (§4).
- We show that methods based on the Generative Pre-trained Transformer model (GPT) present a strong baseline for this task (§5).
- We conduct two analyses that shed light on the strengths of GPT and reveal potential avenues for future research (§6).

2 Implicit Reference Resolution

Task definition. We formally define the task of resolving implicit references as a generation task that requires the prediction of a reference S, given:

1. The original/revised sentence and its preceding context C_p, which includes at least one mention that co-refers to the correct reference (for the example shown in Table 1: *Place the ground beef in a microwave. Microwave until it finishes thawing. Use* _____).
2. The number of tokens L of the reference to be generated according to the final version of a sentence (in case of the example: 2).
3. The follow-up context C_f, which contains the remaining tokens of the original/revised sentence to ensure that the reference fits into the sentence grammatically (in the example, *within 1 or 2 days* needs to fit after *Use* ____).

Performing this task requires a model to generate the sequence of tokens $s_1 \dots s_L$ for the reference S conditioned on the context $\langle C_p, C_f \rangle$. In practice, the full task can be approached by first sampling candidate reference tokens $r_1 \dots r_L$ from a conditional probability distribution $P(r_i|C_p, r_1 \dots r_{i-1})$ and then re-ranking the highest scoring candidates according to the full sequence probability $P(C_p, r_1 \dots r_L, C_f)$. Formulating the task in this way enables a direct application of language models and we demonstrate suitable baselines based on an auto-regressive language model in Section 5.

3 Related Work

The task of resolving implicit references can be viewed as a modified version of implicit argument labeling. First studies on implicit argument labeling were conducted by Gerber and Chai (2010) and Ruppenhofer et al. (2009). Gerber and Chai (2010) collected a dataset by manually labeling implicit arguments of 10 different nominal predicates in NomBank (Meyers et al., 2004), yielding about 1,000 instances. Ruppenhofer et al. (2009) created a dataset through manual annotation of fictional text. Their dataset contains more different predicates than previous studies, but is smaller in size. More recent studies make use of the two datasets and attempted to create additional training data artificially (Silberer and Frank, 2012; Roth and Frank, 2013; Laparra and Rigau, 2013a,b; Chiarcos and Schenk, 2015). Many of them are based on co-reference and discourse salience, which we also use for our baselines. Schenk and Chiarcos (2016) propose an unsupervised approach by aligning implicit arguments to semantic role labeling annotated data. Cheng and Erk (2019, 2018) generated large amounts of training data automatically using co-reference resolution. They also build a neural model based on argument fillers that occur multiple times in a narrative event chain. Finally, there are also datasets with domain-specific annotations such as geographic-event roles (Ebner et al., 2020) and on recipes (Jiang et al., 2020).

Another closely related task is zero anaphora resolution, which has been extensively studied in pro-drop languages such as Chinese (Yeh and Chen,

[1] http://www.wikihow.org

2003) and Japanese (Taira et al., 2008; Isozaki and Hirao, 2003; Seki et al., 2002; Nakaiwa, 1997; Imamura et al., 2009). A closely related study to ours is Imamura et al. (2009), who used language model probabilities as features.

As a commonality, previous work addresses semantic arguments of predicates that are realized outside a local syntactic scope. Our definition of implicit references subsumes such arguments, with the main difference that our task does not require the type of an argument or its semantic role to be specified. As a consequence, references in our task can fill one, none or multiple roles of different predicates. Once the correct reference has been identified, our task additionally requires the generation of a referring expression. This task has been addressed separately in previous work, for instance, using rule-based approaches (Reiter and Dale, 2000), feature-based machine learning (Nenkova and McKeown, 2003; Greenbacker and McCoy, 2009; Same and van Deemter, 2020; Kibrik et al., 2016), and deep neural networks (Castro Ferreira et al., 2016; Cao and Cheung, 2019).

4 Data

The starting point for our data are revision histories from wikiHow, in which we can find insertions of references that were implicit in earlier versions of a sentence. We use wikiHowToImprove (Anthonio et al., 2020), a resource derived from wikiHow that consists of approximately 2.7 million sentences and their revisions. For our purpose, we extract sentences in which a reference was inserted during revision. Most of the sentences in wikiHow are only edited once (about 83%). In other cases, intermediate versions are mostly the result of stylistic refinements or typo corrections. Therefore, we only make use of the final version of a sentence (henceforth *revised sentence*), which includes an inserted reference, and the *original sentence*, in which the reference is assumed to be implicit. As a result, each data point in our collection consists of a pair of two versions of a sentence, henceforth *original-revised sentence pair*. We describe our selection of implicit references in Section 4.1 and present the data statistics in Section 4.2.

4.1 Collection

In order to find pairs with an implicit reference in the original sentence that is explicit in the revised sentence, we take the instances where the revised sentence was created by inserting a word or contiguous set of words in the original sentence. In other words, eliminating the insertion from the revised sentence yields the original sentence. This is a logical starting point, as the implicit reference in the original sentence can be verbalized through insertion. We find cases with contiguous insertions in wikiHowToImprove by computing the differences between the original and revised sentence using `difflib`.[2] As a result, we found 336,129 sentence pairs in which the original sentence was only modified by a contiguous insertion.

In the next step, we identify the subset of insertions that are referential and resolvable in context: that is, we identify words and phrases that refer to a discourse entity. Our study focuses on insertions of single references (i.e., referring expressions that refer to exactly one discourse entity), which are usually not verbalized by sequences exceeding three tokens. Therefore, we only consider insertions that consist of one, two or three word tokens (i.e., unigram, bigram and trigram insertion). We identify references by obtaining co-reference chains on the paragraph level using the Stanza[3] coreference parser. More specifically, by using a combination of the revised sentence and the original context, we can identify referring expressions that are explicit in the revised sentence and coreferent with discourse entities in the original context. Therefore, we parse the revised sentence and the preceding sentences from the *original context*, within the same paragraph.

We add the corresponding original-revised sentence pair to our collection if the full span of the insertion (*full insertion*) or parts of it refer to an entity in the discourse context. In other words, the insertion can contain tokens in addition to the referring expression. However, we only keep insertions that include additional tokens if the additional tokens are required grammatically, given their position in the sentence (e.g., *of you*, *of the shoe*). In particular, we keep the insertions that consist of a reference and specific types of function words (determiners, prepositions) or punctuation.[4] We excluded cases with conjunctions and non-function words as these insertions mainly add or extend factual information. Examples of different insertions

[2] https://docs.python.org/3/library/difflib.html

[3] https://stanfordnlp.github.io/stanza/

[4] We rely on automatic part-of-speech tags for this additional filtering procedure.

Insertion	Reference	Example
unigram (N=2,599)	unigram	**This treatment** can be performed by a dermatologist but **it** is quite expensive.
bigram (N=1,837)	unigram (N=700)	If **you** are using the mobile app, tap the "More" button and then tap **your** name. Select the photo's of **you** tab.
	bigram (N=1,137)	It's not pleasant to read **a book** that has been "personalized" by someone else. If it rains or **the book** gets lost, you'll have to pay to replace **it**.
trigram (N=1,578)	unigram (N=118)	Bend your left **knee** and lift it (as close as you can get **it**).
	bigram (N=1,370)	1. Clean canvas **shoes** by spot washing using a mild detergent and soft toothbrush. Test the spray on the tongue of **the shoe** to make sure it won't stain.
	trigram (N=90)	Check the outer labelling on **the ham shank** to see if **its** fully cooked. If **it** isn't, use the other method instead. Remove the wrapping and place **the ham shank** in a roasting pan.

Table 2: Examples of from our dataset: underlined tokens mark an insertion, tokens in bold highlight references to the same entity. Tokens that are underlined *and* highlighted are the reference tokens to be predicted in our task. Note that the span of the reference can differ from the insertion because of additional tokens (e.g., punctuation).

and inserted references are shown in Table 2. Note that some sentences contain grammar/spelling related errors, which were not corrected in the shown versions of the wikiHow articles.

4.2 Statistics

In total, our collection procedure yields 6,014 instances. More specifically, it contains 2,599 unigram (43.22%), 1,837 bigram (30.50%) and 1,578 trigram (26.24%) insertions. Table 2 shows examples of references and insertions and how they are distributed over different lengths. The numbers indicate that a majority of references are unigrams ($N = 3,854$) and that only a small proportion are trigrams. The table also shows that most references consist of the full insertion (56%), which are 2,599 unigrams, 700 bigrams and 90 trigrams.

Figure 1 indicates the positions of the closest antecedent to resolve an implicit reference. The distribution shows that most references refer to an entity in the same sentence (46.33%, $N = 2,786$) or to an entity in the previous sentence (25.21%, $N = 1,517$). The remaining instances can be resolved within 3 up to 75 sentences. Finally, we observe that in the majority of the 6,014 instances, the reference is mentioned only once (43.15%, $N = 2,595$), twice (18.12%, $N = 1,090$) or three times (9.38%, $N = 564$) in the original context.

In the remainder of this paper, we conduct experiments with our collection of 6,014 implicit

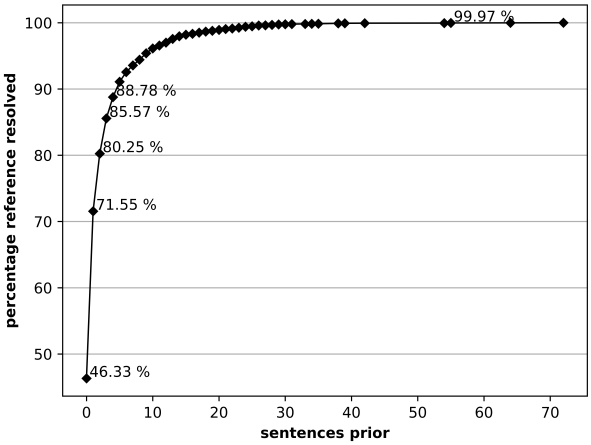

Figure 1: The likelihood that a reference can be found within the previous x sentences in the original context.

references, which we split into a train (81.09%, $N = 4,877$), development (9.83%, $N = 591$) and test (9.08%, $N = 546$) set, following the original split by article of wikiHowToImprove.[5]

5 Language Model Experiments

In this section, we describe a set of experiments in which we investigate the use of a transformer-based language model for the task of resolving implicit references. In particular, we aim to answer the

[5] https://github.com/irshadbhat/wikiHowToImprove

following research questions: Can we find the manually inserted reference among the top completions predicted by a language model, and is it possible to select a correct prediction based on its fit in the sentence or paragraph context? We describe our experimental set-up in Section 5.1 and our results in Section 5.2. Further analyses are provided in Section 6.

5.1 Experimental setting

Data. The starting point for our investigation are the 6,014 instances of original–revised sentence pairs described in Section 4. Each revision involves the insertion of a reference into the given sentence. That is, the revised sentence always contains a reference that was not explicitly present in the original sentence. The full insertion may consist of one, two or three tokens, and it may contain function words in addition to the reference itself.

Method. Resolving implicit references of varying length requires a generative model. We chose the Generative Pre-trained Transformer model (**GPT**) described by Radford et al. (2018) as a benchmark model because it fulfils this requirement and because it is pre-trained on data that does not overlap with our development and test sets.[6] Since GPT is an auto-regressive language model, which means that predictions are made word-by-word (unidirectional), we apply an additional re-ranking procedure over the top-100 generated sequences and their full (left and right) context. For re-ranking, we use the same GPT model and compute its perplexity score for whole sequences on two levels of context: (a) full sentence (**+S-perplexity**) and (b) sentence plus preceding paragraph context (**+P-perplexity**). Finally, we also fine-tune the GPT model on our training set to improve its fit to how-to guides (**+fine-tuning**).

Upper bound and baselines. We approximate an upper bound on our data by assessing the performance of a model that has access to the inserted reference itself, namely the coreference parser used during data creation (see Section 4.1). We further compare GPT to the following baselines: **MostFrequent** always selects the most frequent referring expression of the most frequent entity in the context, **ClosestRef** selects referring expression(s) from the preceding context by how close they are to the point of the insertion, and **TF-IDF** ranks possible n-grams (where n equals the number of tokens in the manually inserted reference) by their tf-idf score (Jones, 1972), for which we take into account all training and development documents.

Evaluation. We evaluate each model by its ability to generate the tokens that are part of the reference inserted in the revised version of a sentence.[7] We count a generated reference as correct if all generated tokens match the tokens in the human-produced reference. To allow for minor variations in spelling, we ignore case when measuring recall (i.e., the relative number of correctly retrieved references) among the top-1 ($R@1$), top-10 ($R@10$) and top-100 ($R@100$) generated sequences.

5.2 Results

We first address our initial question: Can we find manually inserted references among the top completions predicted by a language model? The scores listed in Table 3 indicate the proportion of exact matches that are found within the top-1, top-10 and top-100 references generated by the pre-trained GPT model. The numbers show a similar performance of GPT on the development and test set: In about 37% of the cases, the first-best generated reference is identical to the manually inserted reference. In about 83% of the cases, the manually inserted reference can be found within the top-100 generated references. This result is close to our approximated upper bound: in a random sample from the development set, we found a coreference model, which has access to the manually inserted reference itself, to predict the correct co-reference chain in 86 out of 100 cases (for details, see Appendix A in the supplementary material).

Model comparison. We next attempt to answer our second question, namely is it possible to select the correct reference based on its fit in the sentence or paragraph context? We evaluate two additional steps for selecting references based on the top sequences generated by GPT: model fine-tuning and re-ranking based on sentence-level or paragraph-level perplexity. Table 4 shows the results of each selection approach, combinations and baselines.

[6] Some models, such as GPT-2, were pre-trained on data that includes wikiHow, which could make it possible for them to make correct predictions in our data based on training memory. However, we also experimented with other models in a preliminary study (e.g., XLNet (Yang et al., 2020), TransformerXL (Dai et al., 2019) and BART (Lewis et al., 2019)) and did not observe any advantages over GPT.

[7] Note that a model only needs to generate the reference part of an insertion. Additional words, as described in Section 4, are provided to all models as part of the context.

dataset	R@1	R@10	R@100
develop	37.06% (219 TPs)	67.85% (401 TPs)	82.91% (490 TPs)
test	36.36% (198 TPs)	71.61% (391 TPs)	83.89% (458 TPs)

Table 3: Relative and absolute number of exact matches among the top sequences generated by the GPT model and the manually inserted reference found in a revised sentence.

We observe that GPT substantially outperforms all three baselines. Combining GPT with fine-tuning and paragraph-level perplexity re-ranking leads to an accurate top-1 prediction of the inserted reference in 57.4% of the cases on the test set. In 80.8% of the cases, the inserted reference can be found within the top-10 re-ranked sequences. In ablation experiments on the development set, we find that a combination of fine-tuning and perplexity-based re-ranking is necessary to achieve such high results. Fine-tuning and re-ranking based on sentence-level perplexity only improve R@1 by 2.5 to 3 percentage points, respectively. Without fine-tuning, re-ranking on the sentence level even reduces the chance of finding the correct reference within the top-10 sequences (R@10). Only re-ranking on the paragraph level consistently improves results, up to 8.2 and 16.8 percentage points in R@10 and R@1, respectively.

Discussion. We qualitatively analyzed the top-1 predictions of each method on the development set and observed the following trends: re-ranking on the sentence level generally helps in selecting grammatically suitable candidates when the top generated sequences by the original or fine-tuned model does not fit syntactically, for example, due to number or case disagreements. Fine-tuning GPT seems to adapt the scoring of generated references to better match their occurrences in how-to guides: for example, the pronouns *you* and *them* are more frequent in this genre than *I* and *we*. Finally, we observe that re-ranking on the paragraph level considerably improves the selection of noun phrases that resemble references to entities in the discourse. Whereas the sentence-level method often produces generic references (underlined) that make sense superficially (e.g., *Clean off the surface of the glass*), top candidates in the paragraph-level method plausibly fit also in the specific context (e.g., *Clean off the surface of your typewriter*). We discuss the top predictions of both re-ranking methods in more detail in Section 6.

method	dataset	R@1	R@10
Upper bound	sample	86.0%	
MostFrequent	develop	28.8%	
ClosestRef		16.9 %	59.9 %
TF-IDF		11.0 %	30.8 %
GPT		37.1%	67.9%
+**fine-tuning** (FT)		39.6%	73.4%
+**S-perplexity**		40.1%	63.8%
+**P-perplexity**		52.7%	76.1%
+**FT**+**S-perp.**		46.2%	67.3%
+**FT**+**P-perp.**		**56.4%**	**78.0%**
GPT+**FT**+**P-perp.**	test	57.4%	80.8%

Table 4: Results of re-ranking the top-100 generated predictions by the GPT model in terms of recall (relative number of retrieved references); S-perplexity and P-perplexity indicate the application of re-ranking based on GPT's perplexity scores on the full sentence and paragraph, respectively.

6 Analysis

In this section, we aim to answer two questions evoked by the results in Section 5. First, we ask how fine-tuning and perplexity-based re-ranking improved the scoring of the top-100 generated sequences and what differences can be seen among the re-ranked top-10 sequences (Section 6.1). Secondly, we investigate the plausibility of the two highest ranked fillers generated by the model (Section 6.2). The latter analysis provides us with insights regarding the existence of a single most plausible filler (or whether none/two fillers can be plausible) given the discourse context. In cases where the human-inserted reference is among the top-2, the analysis also makes it possible for us to find if/when the human-inserted reference is identified as the most plausible.

6.1 Sentence vs. Paragraph Perplexity

We compare the generated top-10 sequences of the fine-tuned GPT model and the re-ranked variants on the development set. Predictions for three example sentences are shown in Table 5. The examples indicate that sequences generated by the fine-tuned GPT model often lead to ungrammatical sentences (highlighted in italics). For the re-ranked variants,

we observe that such sequences are scored lower and no longer appear among the top ranks.

The examples based on sentence-level re-ranking reveal two unfortunate side-effects: The first is that non-referential candidates may end up in higher positions, simply because they lead to a grammatical sequence. This is particularly visible in Example (2), where the correct sequence was generated by GPT+FT, but ended up outside the top-10 when using GPT+FT+S-perplexity. The second caveat is that sentence-level perplexity increases the rank of entities that are plausible within the sentence but unrelated to the activity described in the article. This is especially visible in Example (3) in Table 5: the phrases *the number*, *the office*, *a friend*, *the person* all seem reasonable candidates in the context of *calling someone*, but none of them directly correspond to *the salon* mentioned in the article. The same applies to Example (1), in which the context mentioned a *container* but contains no references to a *pot*, *bowl*, *cage* or *bag*. It seems that these candidates simply mimic the usage of common knowledge, especially because none of the candidates occur in the preceding context. The aforementioned reasons could explain why sentence-level perplexity without fine-tuning decreased the recall of the manually-inserted reference among the top-10 candidates (see Section 5).

A final interesting observation is that the caveats caused by the sentence-based re-ranking are less present when applying paragraph-level perplexity. After re-ranking on the paragraph level, we find many of the top candidates to be either repetitions of words and phrases from the context or to be closely related to the manually-inserted sequence. This is illustrated by all examples in Table 5. A quantitative analysis further confirmed this insight: based on paragraph-level perplexity, over 20% of the top-10 ranked bigrams and trigrams appear literally in the preceding context, compared to 11% when using sentence-level perplexity.

6.2 Plausibility of Generated Fillers

In the first analysis, we discussed the impact of re-ranking the top-100 sequences generated by the fine-tuned model. However, even the best re-ranking procedure is insufficient to avoid errors in our approach when the manually inserted reference from a revised sentence does not appear among the generated sequences. In this section, we address two questions. First, we investigate whether the human-inserted reference is always the one that best fits the sentence given the context. Secondly, we assess the plausibility of the top-2 completions by the model in case both are different from the human-inserted reference. These questions are motivated by the results from an internal analysis, in which we found a few instances where an annotator preferred a model-generated reference over the human-inserted reference or had no preference between the two options (a full report is provided in the supplementary material, Appendix B).

Set-up. We take 100 instances from the development set with their top-2 completions provided by the fine-tuned model and ask a student assistant with a background in Computational Linguistics to provide annotations. We randomly select 50 instances where the human-inserted reference is identical with the best generated sequence ("human-insertion among top-2") and 50 where this is not the case ("human-insertion not among top-2"). We show an annotator two versions of the sentence in randomized order: one with the highest ranked sequence and one with the second-highest ranked sequence generated by the model. We show each version together with the preceding sentences from the paragraph and highlight the generated reference. We ask the annotator to indicate the sentence that fits the context better, whether they both fit or whether neither fits. We discuss the main findings of this experiment below, and provide additional examples in the supplementary material (Appendix C). In the examples below, we underline the human-inserted reference.

Human-insertion among top-2 ($N = 50$). The annotator indicated a strong preference for the human-inserted reference in most cases ($N = 34$). However, there were also cases in which the annotator indicated no preference ($N = 13$). This is likely the case because many generated sequences involved paraphrases of the human-inserted reference in the given context (e.g., *the button/this button*, *the party/your party*). The same holds for the remaining instances ($N = 3$), in which the annotator preferred the other generated sequence over the human-inserted reference (e.g., *the sequence/the process*). In these cases, it seems like annotations simply reflect personal preferences. Finally, we found no cases in which the annotator indicated that neither insertion was fitting.

correct sequence	GPT+FT	GPT+FT+S-perplexity	GPT+FT+P-perplexity
(1) Put the grasshoppers in **the container**.	**the container**, *it and*, a container, the bottom, the lid, it. , *it,* , a plastic, a large, the plastic	the water, the pot, the basket, a bowl, the cage, a bag, a pot, the bag, the refrigerator, a jar	**the container**,. 5, the lid, a container, the bag, the bottom, the box, the water, a bag, the hole
(2) Rinse **the parts** before assembling.	**the parts**, *out the*, *off the*, them off, and dry, your ke, them thoroughly, *them with*, them in, them out	them out, them thoroughly, it out, them off, it off, the area, and rinse, each piece, and dry, each section	**the parts**, them thoroughly, them off, them out, all parts, each part, the components, the pipes, both parts, and dry
(3) Call **the salon** and ask questions	**the salon**, your salon, a salon, a local, the sal, a sal, up the, a hair, the hair, *them and*	the number, the office, a friend, this number, them up, the person, a number, the store, the owner, them in	**the salon**, your salon, a salon, their website, the spa, the stylist, them in, your stylist, each salon, a stylist

Table 5: The top-10 predictions for the fine-tuned GPT model and the reranked predictions using sentence-level and paragraph-level perplexity. Bold sequences represent the correct sequence, italic sequences are ungrammatical in context, and underlined sequences are references to entities that are not mentioned in the context.

Human-insertion not among top-2 ($N = 50$). In a majority of these cases, we found the annotator to select one of the model-generated sequences as best fitting, confirming that completions other than the human-inserted reference can be plausible ($N = 29$). A number of the selected sequences are paraphrases ($N = 11$) of the human-inserted reference, such as: *the form/the application*, *the school/this school* or semantically related sequences ($N = 8$) that differ from the human-inserted reference in terms of specificity (e.g., *let the paint/the nails dry*, *wipe and dry the wood/the floors*, *remove the pan/the vegetables from the heat*, *your laptop/your mac*).

The remaining instances ($N = 10$) involve generated sequences that the annotator indicated as best fitting, although they are incompatible with the human-inserted reference (e.g., *Islam/Christianity*, *the microwave/the freezer*). We take these findings as an indicator that implicit references can be resolved incorrectly and therefore lead to misunderstandings, which could be modelled and anticipated by a language model. Finally, the annotator judged both top-2 sequences to be fitting in 17 cases and none to be fitting in 4 cases. In case of the latter, the completions led to an ungrammatical sentence.

7 Conclusions

In this paper, we introduced the task of resolving implicit references in instructional texts, which might be problematic for readers without prior knowledge of the instructed task. We approached the resolution of implicit references as a generation problem for which we leveraged original-revised sentence pairs from wikiHow. The considered pairs contained an explicit reference in the revised sentence which was non-verbalized in the original sentence. Our dataset is one of the largest datasets with implicit references and contains texts from the multiple different domains covered in wikiHow.

We showed that a pre-trained language model is capable of predicting the human-produced insertion in a majority of cases. The best-performing method, which combines a fine-tuned GPT model and perplexity-based re-ranking, achieved results up to 57.4% (top-1) and 80.8% (top-10). Even without fine-tuning and re-ranking, 71.6% of the human-inserted references appeared in the top-10.

We found sentence-level re-ranking useful to eliminate generated sequences that cause ungrammatical sentences and paragraph-level re-ranking to prioritize sequences that also occur in the preceding discourse. Our analysis revealed that the human-inserted reference is commonly found to fit the discourse better than a model-generated alter-

native. However, we also found cases where other completions were plausible. In the future, we will extend this study and take it as a starting point for examining potential sources of misunderstanding.

Acknowledgements

The research presented in this paper was funded by the DFG Emmy Noether program (RO 4848/2-1).

References

Talita Anthonio, Irshad Bhat, and Michael Roth. 2020. wikiHowToImprove: A resource and analyses on edits in instructional texts. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5721–5729, Marseille, France. European Language Resources Association.

Talita Anthonio and Michael Roth. 2020. What can we learn from noun substitutions in revision histories? In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1359–1370, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Meng Cao and Jackie Chi Kit Cheung. 2019. Referring expression generation using entity profiles. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3163–3172, Hong Kong, China. Association for Computational Linguistics.

Thiago Castro Ferreira, Emiel Krahmer, and Sander Wubben. 2016. Towards more variation in text generation: Developing and evaluating variation models for choice of referential form. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 568–577, Berlin, Germany. Association for Computational Linguistics.

Pengxiang Cheng and Katrin Erk. 2018. Implicit argument prediction with event knowledge. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 831–840, New Orleans, Louisiana. Association for Computational Linguistics.

Pengxiang Cheng and Katrin Erk. 2019. Implicit argument prediction as reading comprehension. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6284–6291.

Christian Chiarcos and Niko Schenk. 2015. Memory-based acquisition of argument structures and its application to implicit role detection. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 178–187, Prague, Czech Republic. Association for Computational Linguistics.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context.

Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2020. Multi-sentence argument linking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8057–8077, Online. Association for Computational Linguistics.

Matthew Gerber and J. Y. Chai. 2012. Semantic role labeling of implicit arguments for nominal predicates. *Computational Linguistics*, 38:755–798.

Matthew Gerber and Joyce Chai. 2010. Beyond NomBank: A study of implicit arguments for nominal predicates. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1583–1592, Uppsala, Sweden. Association for Computational Linguistics.

Charles Greenbacker and Kathleen McCoy. 2009. UDel: Generating referring expressions guided by psycholinguistc findings. In *Proceedings of the 2009 Workshop on Language Generation and Summarisation (UCNLG+Sum 2009)*, pages 101–102, Suntec, Singapore. Association for Computational Linguistics.

Kenji Imamura, Kuniko Saito, and Tomoko Izumi. 2009. Discriminative approach to predicate-argument structure analysis with zero-anaphora resolution. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 85–88, Suntec, Singapore. Association for Computational Linguistics.

Hideki Isozaki and Tsutomu Hirao. 2003. Japanese zero pronoun resolution based on ranking rules and machine learning. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 184–191.

Yiwei Jiang, Klim Zaporojets, Johannes Deleu, Thomas Demeester, and Chris Develder. 2020. Recipe instruction semantics corpus (RISeC): Resolving semantic structure and zero anaphora in recipes. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 821–826, Suzhou, China. Association for Computational Linguistics.

Karen Spärck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21.

Andrej A. Kibrik, Mariya V. Khudyakova, Grigory B. Dobrov, Anastasia Linnik, and Dmitrij A. Zalmanov. 2016. Referential choice: Predictability and its limits. *Frontiers in Psychology*, 7(1429).

Egoitz Laparra and German Rigau. 2013a. ImpAr: A deterministic algorithm for implicit semantic role labelling. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1180–1189, Sofia, Bulgaria. Association for Computational Linguistics.

Egoitz Laparra and German Rigau. 2013b. Sources of evidence for implicit argument resolution. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Long Papers*, pages 155–166, Potsdam, Germany. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004. The NomBank project: An interim report. In *Proceedings of the Workshop Frontiers in Corpus Annotation at HLT-NAACL 2004*, pages 24–31, Boston, Massachusetts, USA. Association for Computational Linguistics.

Tatjana Moor, Michael Roth, and Anette Frank. 2013. Predicate-specific annotations for implicit role binding: Corpus annotation, data analysis and evaluation experiments. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Short Papers*, pages 369–375, Potsdam, Germany. Association for Computational Linguistics.

Hiromi Nakaiwa. 1997. Automatic extraction of rules for anaphora resolution of japanese zero pronouns from aligned sentence pairs. In *Proceedings of a Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*, ANARESOLUTION '97, page 22–29, USA. Association for Computational Linguistics.

Ani Nenkova and Kathleen McKeown. 2003. References to named entities: a corpus study. In *Companion Volume of the Proceedings of HLT-NAACL 2003 - Short Papers*, pages 70–72.

Alec Radford, Karthik Narasimhan, Tim salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. OpenAI.

Ehud Reiter and Robert Dale. 2000. *Building Natural Language Generation Systems*. Studies in Natural Language Processing. Cambridge University Press.

Michael Roth and Anette Frank. 2013. Automatically identifying implicit arguments to improve argument linking and coherence modeling. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 306–316, Atlanta, Georgia, USA. Association for Computational Linguistics.

Josef Ruppenhofer, Caroline Sporleder, Roser Morante, Collin Baker, and Martha Palmer. 2009. SemEval-2010 task 10: Linking events and their participants in discourse. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2009)*, pages 106–111, Boulder, Colorado. Association for Computational Linguistics.

Fahime Same and Kees van Deemter. 2020. A linguistic perspective on reference: Choosing a feature set for generating referring expressions in context. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4575–4586, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Niko Schenk and Christian Chiarcos. 2016. Unsupervised learning of prototypical fillers for implicit semantic role labeling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1473–1479, San Diego, California. Association for Computational Linguistics.

Kazuhiro Seki, Atsushi Fujii, and Tetsuya Ishikawa. 2002. A probabilistic method for analyzing Japanese anaphora integrating zero pronoun detection and resolution. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Carina Silberer and Anette Frank. 2012. Casting implicit role linking as an anaphora resolution task. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 1–10, Montréal, Canada. Association for Computational Linguistics.

Hirotoshi Taira, Sanae Fujita, and Masaaki Nagata. 2008. A Japanese predicate argument structure analysis using decision lists. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 523–532, Honolulu, Hawaii. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2020. Xlnet: Generalized autoregressive pretraining for language understanding.

Ching-Long Yeh and Yi-Chun Chen. 2003. Using zero anaphora resolution to improve text categorization. In *Proceedings of the 17th Pacific Asia Conference on Language, Information and Computation*, pages 423–430, Sentosa, Singapore. COLIPS PUBLICATIONS.

A Co-reference Quality Analysis

Human-inserted reference co-refers
1. Take your pregnant cat to the vet . As soon as you know , or suspect , that your cat might be pregnant , you should take **her** to the vets to get **her** checked over .
1. On the sheet of **foam**, draw an outline like the one in the image. Make sure it can wrap around your bottle around once, and the strip pointing down can fold under the bottom of the bottle. 2. Cut the foam shape out. 3. Cut a piece of duct tape off the roll, so it will cover part of the bottom of **the foam** and about on each side. Take a new piece and start rolling it around the sides of **the foam** .

Table 6: Examples where the human-inserted reference co-refers with an entity in the context according to Stanza. The human-inserted reference is highlighted and underlined. The referring expressions within the same co-reference chain are highlighted.

Human-inserted reference does not co-refer
1. Know what kind of games you like (strategy, action, adventure, racing, rpg, simulators, etc. 2. Strategy games are good for you when thinking. They help you reinforce what you learned. **Some strategy games** are : Age Of Empires, The Settlers, City Of Heroes. Action games: **these** are liked by many people.
1. Go get an ' on the floor ' cat scratch pad from anywhere. This one cost $6. My cats wo n't use it flat on the floor, like this. 2. Remove the corrugated cardboard from the box. They usually glue it down, probably because they do n't want you to flip it and re-use the back - side. (You can.) 3. Using the edge of **a counter** or table or other sturdy piece of furniture, break **it** to fit.

Table 7: Examples where the human-inserted reference did not co-refer with the referring expressions within the same co-reference chain according to Stanza. The human-inserted reference is highlighted and underlined, whereas the referring expressions within the same co-reference chain are highlighted.

In this section, we describe a study that we conducted to investigate the quality of the obtained co-reference chains from Stanza.

Method. To investigate the quality of the co-reference chains, we asked an annotator to identify whether the human-inserted reference occurred earlier in the context. We specifically showed the annotator the context and the revised sentence, in which we marked the human-inserted referring expression. We additionally highlighted the referring expressions that co-referred with the human-inserted reference. We asked the annotator to annotate 100 instances, which we randomly extracted from the development set of our data (see Section 4). The annotator was a student in a Computational Linguistics program.

Results. The annotator found 86 instances where the human-produced reference co-referred with the referring expressions indicated by the Stanza co-reference parser. We show two examples of these instances in Table 6. In the remaining instances ($N = 14$), the human-inserted referring expression did not co-refer with the referring expressions indicated by Stanza. Nonetheless, it is still possible to find a suitable antecedent in such cases, as shown by the examples in Table 7. We conclude from the obtained results that implicit references in our dataset (described in Section 4) are generally co-referent with an entity mentioned in the context.

B Error Analysis

In Section 6.1, we discussed the impact of re-ranking the top-100 sequences generated by the fine-tuned GPT model. However, even the best re-ranking procedure is insufficient to avoid errors in our approach when the manually inserted reference from a revised sentence does not appear among the generated sequences. Therefore, we perform an additional study in which we analyze cases of human-produced references that do not show up among the top-100 candidates generated by the fine-tuned GPT model.

Set-up. We select all instances for which the fine-tuned GPT model did not generate the human-produced reference among the top-100 ($N = 84$) and ask one annotator to provide judgements. We show the annotator two versions of the revised sentence in randomized order: one containing the human-produced reference and the other containing the top-1 generated completion by GPT. Each

version is shown together with the preceding sentences from the paragraph.

In the annotation interface, we highlight the references and ask the annotator to select the version that fits the sentence better, given the context, or whether "both fit". We discuss the three different outcomes below.

Preference for human insertion ($N = 57$). In most cases, the annotator labeled the human-produced reference as being a better fit than the generated sequence. In 26 of these cases, the model-generated sequence was not a reference or the reference was accompanied by function words or punctuation. Both types of sequences usually yield an ungrammatical sentence.

The remaining 31 instances can be categorized into three groups: The first are cases where the generated sequence does not make sense in the given sentence position ($N = 12$), such as *Rub the dog's coat with **the chamois***. Another subset ($N = 13$) consists of sequences that are referential but the insertion yields an ungrammatical sentence, for example: *It can also be unpleasant to withdraw from **your***. The rest ($N = 6$) seem to be sensible references according to our observations, such as:

> (1) Leave your diya uncovered at room temperature, and do your best to keep it away from moisture. The clay should set after 24 hours. If you put your diya on a plate or mat and notice it starts drooping, lightly grease a sheet of aluminum foil with **your spatula/**vegetable oil.

Here, the generated (boldface) sequence *your spatula* seems to fulfil a plausible but different semantic role than the human-produced (underlined) sequence *vegetable oil*. Even though it remains unclear why the annotator preferred the human-produced sequence in these cases, it is interesting to see that the model managed to generate a reference that fills a different semantic role in the given sentence (cf. summary below).

No preference ($N = 20$). In 20 out of 84 instances (23.81%), the annotator marked both sequences as being equally fitting in the context. Some of the produced sequences were not references or entities ($N = 3$), but still plausible insertions in the sentence, for example, *Open or create a **new** word document.* The remaining cases ($N = 17$) contain generated references which are plausible but different from the human-produced reference, such as: *Many of your answers on **the subject/**the regents* ... (the context here is an article on regents exams). The high relative frequency of such cases suggests that the model-generated sequences might be able to reflect alternate, plausible fillers for an implicit reference.

Preference for model insertion ($N = 7$). In 7 cases (8.33%), the annotator marked the generated sequence as fitting better than the human-produced sequence. In 4 cases, the human-produced reference caused fluency or grammatical issues.[8] In two instances, the generated sequence referred to a different entity than the human-produced reference, such as:

> (2) Look closely and carefully at the grain pattern on the handbag. The pattern of the grain on a crocodile leather handbag will have some irregularities. If the grain pattern of **the leather/**the scales is very uniform , it has probably been stamped on .

Given the annotator's preference, these examples support the finding that model-generated sequences may reflect alternate, plausible fillers of an implicit reference. We also noticed one instance where a generated sequence filled the same semantic role as the human-produced reference, but differed in terms of granularity.

> (3) Click the tab at the top left that says "Themes". This will take your sidebar to the theme garden, which looks like this. Make sure to choose from the type of theme you want to look at first from **the sidebar/**wikihow.

In this case, the annotator might have chosen the generated sequence because it was mentioned in the previous sentence.

Summary. We showed that most incorrect predictions of the fine-tuned GPT model are indeed errors, as confirmed by the annotator's preference for the human-produced reference over the generated references. We further found the annotator's preference to reflect the effects of re-ranking when the correct reference can be found among the top-100 candidates: The annotator also preferred references that do not disrupt grammaticality and that

[8] It might be that further revision is needed or that they refer to an external link/image.

also occur in the preceding discourse. Finally, there are a fair number of cases in which the annotator had no preference or preferred the model-generated sequence, indicating that there exist plausible, alternate references. In cases like Example (1), such references can be distinguished by the semantic role they fill. However, we also find examples, such as (2) and (3), in which different references can fill the same role at varying levels of granularity. Therefore, it seems unclear whether semantic roles would be helpful in this task and what a suitable role inventory would be.

C Additional Examples for Section 6.2

In this section, we provide additional examples for the analysis conducted in Section 6.2. In the examples, we highlight the preferred filler by the annotator and underline the human-inserted reference.

C.1 Human-insertion among top-2

An example where the annotator preferred the human-insertion is:

> (1) 1. Read books or go to the Library. Kids love it when you take them there. 2. Play games with them. Little kids like games like 'Simon says', 'Hide and seek', 'Tag', etc. Older kids might play board games or video games. 3. Make up your own games. Kids have a great time doing this ! Watch a movie with friends/**them**.

In this example, the annotator probably preferred the human-inserted reference because the top-2 completion *friends* does not make sense in the given context. Instead, the referring expression *them* seems more plausible because it can be used to refer to *children*.

In addition, we show an example where the annotator had no preference below:

> (2) Has a friend of yours read something personal or embarrassing that belongs to you? Here 's some tips on how to deal with **that/it**.

Example (2) shows a common phenomenon that we noticed for all the instances where the annotator had no preference, namely that the fillers are paraphrases in the given context.

Finally, we show two examples where the annotator preferred the top-2 generated completion by the model, instead of the human-insertion.

> (3) The corks will retain moisture longer than traditional milch and help maintain your plant's health between waterings. 2. Use as a fire starter. When you need to start a fire, remove a cork or two and place them under the wood to be kindled before lighting **the fire**/a fire.

> (4) it's as if you're trying to brush some debris off your pants. Return to the original position. repeat the process with your left knee. 5. Practice the two - step. the two - step is a very basic dance move that can help you get into the rhythm of the music. practicing the two - step can help you form a dance routine. Repeat **the process**/the sequence with your left food.

Both examples are instances for which we concluded that the annotations reflect personal preferences, since the human-insertion and top-2 filler are paraphrases.

C.2 Human-insertion not among top-2

Annother has preference. The two examples below are instances where the annotator preferred either of the top-2 generated sequences.

> (5) Use the pie within several months. While a properly frozen pecan pie will last for a while in a freezer, it won't last forever. Try to use the pie within 2 months, as after that it is at risk of developing freezer burn. * To reheat a frozen pie, let it thaw overnight in the refrigerator. Then warm it in a oven for 15 to 20 minutes. The pie will do better if it is kept at a constant temperature in **the oven**/the microwave.

In example (5), the human-inserted reference was the freezer, which also differs from the top-1 and top-2 completions by the fine-tuned model. This example indicates the possibility of the annotator preferring a different filler than the human-inserted reference and therefore a mismatch between the interpretation of the implicit reference of the writer

and a reader.
The second example shows differences between the top-2 in terms of granularity:

> (6) Wash your face/**your mouth** with warm water in the morning.

In this case, the human-inserted reference was your lips. The annotator therefore preferred the filler that was the closest to the human-inserted reference.

Annotator has no preference. Finally, we show an example from the set where the annotator had no preference. This subset consisted of paraphrases, such as:

> (..) It's recommended that you use the manual setting in order to manipulate the flash to produce the highest quality photos. Change the power of **a flash/the flash** depending on the ambient light and the subject you are shooting .

The high occurrence of paraphrases in the generated fillers shows that GPT can generate several plausible fillers for a given implicit reference and is an interesting point for future research.