SemEval-2022 Task 7: Identifying Plausible Clarifications of Implicit and Underspecified Phrases in Instructional Texts

Michael Roth **Talita Anthonio** **Anna Sauer**
University of Stuttgart
Institute for Natural Language Processing
{michael.roth,talita.anthonio,anna.sauer}@ims.uni-stuttgart.de

Abstract

We describe SemEval-2022 Task 7, a shared task on rating the plausibility of clarifications in English-language instructional texts. The dataset for this task consists of manually clarified how-to guides for which we generated alternative clarifications and collected human plausibility judgements.[1] The task of participating systems was to automatically determine the plausibility of a clarification in the respective context. In total, 21 participants took part in this task, with the best system achieving an accuracy of 68.9%. This report summarizes the results and findings from 8 teams and their system descriptions. Finally, we show in an additional evaluation that predictions by the top participating team make it possible to identify contexts with multiple plausible clarifications with an accuracy of 75.2%.

1 Introduction

Understanding texts in natural language requires that both explicit text components as well as implicit references and relationships are interpreted correctly. This applies in particular to instructional texts, which demand a clear understanding of individual instruction steps in order to reach the desired goal. Possible uncertainties should therefore already be clarified in the text. In principle, such clarifications can also be generated automatically. In that case, however, it will be necessary to investigate the circumstances under which a clarification is plausible and unambiguous.

As a first step towards such an investigation, this shared task evaluates the ability of NLP systems to distinguish between plausible and implausible clarifications of an instruction. Inspired by the success of previous cloze-based evaluations (see Section 2), we set up our task as a cloze task, in which clarifications are presented as fillers and systems have to identify which fillers plausibly fit in a given context (see Table 1). Our focus in this task is on different types of referring expressions that are either underspecified or not realized explicitly at all, and we consider possible clarifications in the form of additional specification or explicitation.

	Choose a Hair Salon		
(1)	Check ratings of different salons.		
(2)	Visit the salon's website.		
(3)	Call $\underline{\quad\emptyset\quad}$ and ask questions.		
✓	the salon	✗	a friend
✓	the number	✗	your stylist
✓	the owner		

Table 1: Simplified example from a pilot study: the top shows a sentence (3) and shortened version of its discourse context (1–2). In the clarified version of this sentence, the phrase *the salon* was inserted. Other phrases shown in the bottom part are automatically generated fillers, annotated as plausible (✓) or implausible (✗).

Research in linguistics and psychology has shown that individuals use language differently (Pennebaker and King, 1999; Heylighen and Dewaele, 2002). In particular when it comes to implicit and underspecified language, individual differences can also lead to different interpretations (Scholman and Demberg, 2017; Poesio et al., 2019). As a result, worst case scenarios include medical instructions being followed incorrectly or news being passed on inaccurately. In view of the fact that language is inherently ambiguous, however, it is neither sensible nor expedient to produce clarifications for all occurrences of underspecification. Avoiding worst-case scenarios therefore goes beyond ranking individual clarifications by plausibility and must take into account whether multiple (incompatible) clarifications are perceived as plausible, thus reflecting possible misunderstandings.

We discuss our task and data in more detail in

[1] The task data is available at https://github.com/acidAnn/claire.

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 1039 - 1049
July 14-15, 2022 ©2022 Association for Computational Linguistics

Sections 3 and 4, respectively. The participating teams are summarized in Section 5 and their results on the task and additional evaluations in Section 6.

2 Related Work

Cloze tasks have become a standard framework for evaluating various discourse-level phenomena in NLP. Some prominent examples include the narrative cloze test (Chambers and Jurafsky, 2008), the story cloze test (Mostafazadeh et al., 2016), and the LAMBADA word prediction task (Paperno et al., 2016). In these tasks, NLP systems are required to make a prediction about the filler of a cloze that is most likely to continue the discourse. However, it is not always clear whether exactly one likely filler exists. Evaluations typically circumvent this issue by requiring systems only to distinguish between a correct and an incorrect filler, or by evaluating predictions only with a relative measure. Both of these options ignore the more general challenge that multiple fillers can be plausible. Our shared task addresses this challenge explicitly, by requiring systems to classify different clarifying fillers as either plausible or implausible. This is a natural extension of previous cloze tasks to discourse contexts in which multiple interpretations are plausible. This extension makes it possible to evaluate in how far NLP systems can reflect cases of underspecification and uncertainty as well as possible sources of misunderstanding.

Our task is based on manual text revisions that can be traced through revision histories and by which possible needs for clarification can be identified. Thus, the task follows a number of existing research contributions that deal with text revisions. A number of previous works examine reasons for and types of revisions in, for example, Wikipedia (Bronner and Monz, 2012; Daxenberger and Gurevych, 2012; Yang et al., 2017; Faruqui et al., 2018) and the essay-based corpus ArgRewrite (Zhang and Litman, 2015, 2016; Afrin and Litman, 2019; Kashefi et al., 2022). In this work, we use revision histories of instructional texts because clarifications seem particularly relevant in this domain.

The starting point of our task is the data set wikiHowToImprove (Anthonio et al., 2020), which comprises revision histories of more than 250,000 how-to guides from the online platform wikiHow.[2] In our own previous work, we investigated the extent to which these histories are useful for examining textual improvements (Anthonio and Roth, 2020), predicting revision requirements (Bhat et al., 2020), modeling cases of lexical vagueness (Debnath and Roth, 2021), and resolving implicit references (Anthonio and Roth, 2021). The last two studies in particular have shown that wikiHowToImprove is well suited as a resource for studying clarifications of semantic phenomena. We describe one of these studies and how we build upon it in more detail in Section 3.

3 Task and Background

The general idea of the present shared task is to use revisions of English-language instructional texts as a basis to identify potential clarifications and to rate them regarding their plausibility. We assume that at least certain cases of clarifying revisions follow patterns that can be recognized automatically, by comparing the text before/after revision. An example of such a pattern is the insertion of a nominal phrase mentioned in context that makes an implicit reference explicit (see Table 1). We consider additional patterns as part of this shared task (see Section 4), but only consider cases of insertion for simplicity.

The focus on insertions allows us to consider clarifications as solutions to a cloze test, since the revision always fills in a text segment that previously was not present. Compared to previous cloze tasks, we do not assume that the revision observed is always unique and plausible. Instead, we also consider alternative clarification options and obtain plausibility judgments for all options.

As background, we first summarize findings from a pilot study that we conducted before setting up the present task (§3.1). Based on this, we then describe the settings of the shared task (§3.2).

3.1 Pilot Study

In our pilot study (Anthonio and Roth, 2021), we constructed a dataset of implicit references and potential clarifications in three steps: (1) heuristically identifying insertions of nominal phrases mentioned in the previous context, (2) automatically generating alternative clarifications using generative language modeling (GPT; Radford et al., 2018), and (3) collecting human plausibility judgements for each clarification option.

The first step of our pilot showed that it is possible to extract about 6,000 relevant clarifications from the revisions in wikiHowToImprove. We fur-

[2] www.wikihow.org

ther found that most noisy instances can be filtered by the application of linguistic constraints and that remaining cases can be identified during the manual verification in the final step. In the second step, we found GPT to produce completions for many sentences that seem sensible on the surface level. Using our best strategy, namely re-ranking based on paragraph-level perplexity, the best sequence generated by the model was identical to the human-inserted clarification in over 56% of cases, and the clarification appeared in the top-10 generated sequences in 78% of cases.

A crucial finding in the third step was that the annotator indicated a preference for the human-inserted clarification in most cases (68%), but different, model-generated clarifications were judged as equally good in many cases (24%). In some cases, the annotator actually preferred a generated clarification over the human-edited insertion (8%).

The framework for the shared task is strongly motivated by the finding that alternative clarifications, generated by a computational model, can be as good or even better than human-produced clarifications. In some cases, we simply found different verbalizations of the same proposition. In other cases, like examples (a) and (b) below, we found plausible alternatives that are not fully compatible semantically.

(a) Call <u>the salon</u> and ask questions.

(b) Call <u>each salon</u> and ask questions.

When multiple incompatible readings exist, there is a risk that instructions will be misunderstood and not lead to the desired goal. To identify potential occurrences of such cases, we consider different fillers in the shared task and rate the plausibility of each filler independently.

3.2 Shared Task Settings

The SemEval shared task is set up as follows: Systems are provided with a cloze sentence, surrounding sentences and a potential clarifying filler as input, and are required to make a prediction regarding the plausibility of the filler in the given context. For evaluation, predicted labels are compared against the manually collected plausibility judgements described in Section 4. We define two subtasks with different labels and evaluation measures.

Task 1: Classification. In the classification task, systems need to distinguish between three labels (IMPLAUSIBLE, NEUTRAL and PLAUSIBLE). We use *accuracy* as the main evaluation measure, calculated as the proportion of correct predictions among all predictions of a system.

Task 2: Ranking. In the ranking task, systems need to predict a continuous plausibility score. We evaluate the predictions based on their *correlation* with human judgements, calculated as Spearman's rank correlation coefficient between all predictions and all judgements.

We describe the selection of data and collection of human judgements in the next section. In Section 6, we discuss additional evaluations performed to assess system performance with regard to the presence of multiple plausible clarifications.

4 Data

We closely follow the three steps of our pilot study, described in Section 3.1, to construct the data for this shared task. Our starting point is the dataset wikiHowToImprove (Anthonio et al., 2020), a resource of sentence-level revisions and their contexts based on wikiHow. In the first two steps, we create relevant data from this resource automatically; in the final step, we collect manual annotations to form a gold standard. In step 1, we apply a pattern-based approach to identify revisions that involve insertions that serve specific clarifying functions (§4.1). In step 2, we use transformer-based language models to produce sets of alternate clarifications that may or may not be compatible with an observed insertion (§4.2). In step 3, we collect human plausibility judgements on each clarification independently (§4.3).

4.1 Data Extraction

We collect relevant revisions by identifying cases in which a single *contiguous* insertion and no other change was made within a sentence. We compute differences and extract cases automatically based on the Python library `difflib`[3] and the following preprocessing tools: `spaCy`[4] for sentence splitting and tokenization, the Berkeley Neural Parser (Kitaev and Klein, 2018) for constituency parsing and `Stanza` (Qi et al., 2020) for POS tagging, dependency parsing and co-reference resolution. For the shared task, we focus on four types of phenomena, which are summarized in Table 2.

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

[4] https://github.com/explosion/spaCy

Phenomenon	Clarification pattern	Example	Potential filler
Implicit Reference	$\emptyset \rightarrow$ [DET] NOUN	Lift your toes up while keeping your leg straight. Hold __$\emptyset$__ for a few seconds, then release. Incorporate calf stretches into your yoga routine.	✓ your pose ✓ the stretch ? a leg ? the chair ✗ your head
Fused head	DET/JJ $\emptyset \rightarrow$ DET/JJ NOUN	Traditionally, the groom waits for the bride at the altar, the bride tosses the bouquet and (...). Since this is your wedding, feel free to change these __$\emptyset$__.	✓ ideas ✓ plans ? symbols ? characters ✗ changes
Noun compound	$\emptyset$ NOUN $\rightarrow$ NOUN NOUN	Heating for cold water tanks isn't quite of an issue as for tropicals. In fact, you can keep a __$\emptyset$__ tank without a heater.	✓ goldfish ✓ freshwater ✓ water ? fishing ✗ soup
Metonymy	NP $\emptyset \rightarrow$ NP's NP $\emptyset$ NP $\rightarrow$ NP of NP	Look at the __$\emptyset$__ of the teeth. If you're unsure of your dog's age, or want to determine if they are already entering into the senior territory, try the teeth.	✓ condition ✓ color ✓ thickness ? layout ? points

Table 2: Phenomena, extraction patterns and example clarifications (✓ plausible, ? neutral, ✗ implausible).

Implicit references. Instances with a non-verbalized reference in the original sentence which was clarified in the revised sentence through insertion. We select the cases from Anthonio and Roth (2021) with insertions containing a single noun or a determiner followed by a noun.

Fused heads. Instances of noun phrases for which the head noun was implicit in the original sentence and clarified in the revised sentence through insertion. We search for noun phrases with a determiner or adjective head in the original sentence and select those instances where a single noun was inserted in the revision.

Noun compounds. Instances of underspecified noun phrases, which were clarified in the revised sentence through the insertion of a dependent noun to form a more specific compound. We select instances of single noun insertions in which the inserted noun is a `compound` dependent of another noun that has already been present in the original sentence.

Metonymy. Instances in which a revision adds a noun y to a noun x to make explicit to which component or aspect of x the text refers. For the genitive pattern x*'(s)* y, we select insertions including an apostrophe and a noun y that is in a dependency relation `nmod:poss` with a noun x. For the y *of* x pattern, we select insertions that consist of a noun y and the token *of* added right in front of a noun x, allowing for intervening determiners and adjectives.

4.2 Constructing Clarifications

We produce a set of possible clarifications for each instance as follows: First, we generate the top-100 fillers in place of an observed insertion using language modeling. Second, we select a subset of potentially suitable clarifications by filtering and clustering the top-100.

Filler generation. For the implicit references, we take the top-100 generated clarifications from Anthonio and Roth (2021). For the other phenomena, we generate alternative clarifications automatically using the same approach as Anthonio and Roth (2021). That is, we feed the original sentence s with the surrounding sentences from the same paragraph to a language model. We then compute the top-100 completions for the token position(s) where an insertion was added in the revised sentence. We use BERT (Devlin et al., 2019) instead of GPT (Radford et al., 2018) to generate the clarifications, as the required insertions consist of only one token and BERT makes it possible to also consider follow-up context directly. The BERT checkpoint `bert-base-uncasedbert-base-uncased`[5] in `Transformers` (Wolf et al., 2020) was used without additional pre-training.

Filler selection. From the top-100 clarifications provided by the language model, we select four fillers with the goal of producing a semantically diverse set of clarifications. First, we remove unsuitable fillers from the top-100, including cases that only consist of digits or non-alphanumerical characters and fillers that do not have the right part of speech based on `Stanza` (retaining only *NOUN* for fused heads and metonymy and *NN* for noun compounds to exclude plural nouns).

For all instances with ≥ 4 candidate fillers, we select the observed insertion from the revised sentence as one filler. To select semantically different fillers as alternate candidates, we apply k-means clustering with $k = 4$ to the remaining candidates, using the algorithm by Elkan (2003) as implemented in `sklearn` (Pedregosa et al., 2011). We obtain vector representations for clustering from BERT (`bert-base-uncased`) by averaging over the last hidden state for all tokens in a filler. After clustering, we select the fillers closest to the four cluster centroids based on cosine similarity.

4.3 Plausibility Annotation

Task. After selecting fillers for each sentence, we collect plausibility judgements on Amazon Mechanical Turk for our train set (19,975 instances, i.e. 3995 sentences with 1 human and 4 generated fillers each[6]), development and test sets (2,500 instances each, i.e., 125 sentences per phenomenon with 5 fillers per sentence). Each clarification in the training set is annotated by 2 crowdworkers. For the development and test set, we collected annotations from 4 crowdworkers to ensure a consistently high quality. In each annotation task, we ask participants to indicate on a scale from 1 to 5 whether the clarification made sense in the given how-to-guide. A screenshot of the interface for our Human Intelligence Task (HIT) is provided in Appendix A.

	Train	**Dev**	**Test**
IMPLAUSIBLE	5,474 (27%)	982 (39%)	858 (34%)
NEUTRAL	7,162 (36%)	602 (24%)	672 (27%)
PLAUSIBLE	7,339 (37%)	916 (37%)	970 (39%)
Total	19,975	2,500	2,500

Table 3: Distribution of class labels in our training, development and test sets.

Qualifications. We use several qualifications to increase the annotation quality. First, we require participants to be located in the United States or in the United Kingdom, to increase the chance that the participants are native speakers of English. Secondly, participants need to have a HIT approval rate $\geq 95\%$ and their number of approved HITS has to be ≥ 1000. Finally, annotators are required to pass a qualification test in which they are asked to judge a list of clearly plausible and implausible cases that were pre-selected unanimously by the authors.

Class labels. For Task 1 (classification), we average over the real-valued judgements collected for a clarification and map this plausibility score to one of the three classes labels. Specifically, we label clarifications with an average score ≤ 2.5 as IMPLAUSIBLE, clarifications with a score ≥ 4.0 as PLAUSIBLE, and all clarifications between these thresholds as NEUTRAL. The thresholds have been selected based on manual inspection of the data and mathematical considerations: in particular, the threshold for PLAUSIBLE requires scores to be substantially above average (in case of two judgements, $\geq$3&5 or $\geq$4&4), whereas the IMPLAUSIBLE threshold allows for a slightly wider range of judgements. The NEUTRAL label covers cases that received inconclusive individual scores as well as cases of disagreement (e.g. 3&3 as well as 2&5).

Statistics. We show the frequency distribution of the labels in the train, development and test set in Table 3. It is noteworthy that development and test set proportionally includes fewer NEUTRAL and more IMPLAUSIBLE clarifications than the training set. Presumably, this is because we increased the number of qualification questions from 4 to 6 after collecting the training data to ensure the quality of the evaluation data.

Since we are particularly interested in cases with

[5] We also tried `bert-base-cased` in preliminary experiments but observed no improvements.

[6] 1000 each for noun compounds and metonymy, 996 for implicit references and 999 for fused heads.

Team	Model type	Pre-trained model components	Additional comments
X-PuDu	ensemble	DeBERTa, ERNIE, XLM-R	pattern-aware, multi-loss
HW-TSC	ensemble	DeBERTa, RoBERTa, S-BERT	incl. unsupervised model
PALI	ensemble	DeBERTa, RoBERTa, XLM-R	pattern-aware, multi-loss
Nowruz	Transformer	T5	ordinal regression, multi-loss
JBNU-CCLab	ensemble	DeBERTa	–
DuluthNLP	Transformer	ELECTRA	class weighting
Stanford MLab	Transformer	ELECTRA	–
niksss	Transformer	BERT	–

Table 4: Summary of the best models on the test set according to the submitted system descriptions.

multiple plausible clarifications, we also compute the average number of PLAUSIBLE clarifications per sentence s, which we found to be 1.84, 1.87 and 1.84 in the training, development and test set, respectively. This means that, on average, each annotated sentence in the dataset has between 1 and 2 clarifications that the annotators rated as plausible.

5 Participants

A total of 21 users participated in the CodaLab competition set up for the shared task and 8 teams submitted system description papers. An overview of the best model by each team is shown in Table 4.[7] We observe that all systems are based on Transformer architectures, using one or more of the following pre-trained models: BERT (Devlin et al., 2019), DeBERTa (He et al., 2020), ELECTRA (Clark et al., 2019), ERNIE (Sun et al., 2019), RoBERTa (Liu et al., 2019), S-BERT (Reimers and Gurevych, 2019), T5 (Raffel et al., 2020), XLM-R (Conneau et al., 2020).

In addition to fine-tuning a single or multiple Transformer models in an ensemble, some teams have taken additional steps to adapt their system to the task. We summarize some of these steps below.

Consideration of phenomena. At least two teams took into account that the data set consists of four phenomena that were identified using different patterns (*pattern-aware*): PALI used the phenomenon description that applies to a classification instance as additional model input; X-PuDu developed an ensemble architecture that consists of different individual models and hyperparamters for each phenomenon.

Adapted loss functions. Several teams adapted the loss functions of their models to better account for various properties of the task. This includes the use of classification and regression based loss functions in a multi-task learning set-up (*multi-loss*) as well as the use of specific loss functions that consider the ordinal nature of labels (*ordinal regression*) or differences in label distributions (*class weighting*) in the classification task.

Unsupervised components. Given the similarity of our task to general cloze tasks, several teams experimented with models that were merely self-supervised and not fine-tuned on task-specific training data. In case of one team, HW-TSC, such an unsupervised component is also part of the ensemble model that produced the best results.

6 Results and Discussion

The results for Task 1 and 2 are shown in Table 5 and 6, respectively. We focus our discussion on Task 1: Classification, as the participants of Task 2 form only a subset of the Task 1 participants and the system results rank, with exception of the last two teams, in the same order. In addition to showing results by participants, we also provide a human upper bound as well as results by our own BERT-based baseline model. The upper bound was computed as the accuracy over all individual annotations when compared against the (averaged) class label of each test instance.

The human upper bound has an accuracy of 79.4%, indicating that the task is challenging and potentially involves a number of disagreements. The winning team of the competition, X-PuDu, achieves an accuracy of 68.9%, only 10.5 percentage points below the human upper bound. The results of all teams lies substantially above a naive majority class baseline of 39%. All teams but one

[7] A table with the official results of the CodaLab competition, including participants who did not submit system descriptions, is shown in Appendix B.

Rank	Team	Accuracy
–	Human (upper bound)	79.4%
1	X-PuDu	68.9%
2	HW-TSC	66.1%
3	PALI	65.4%
4	Nowruz	62.4%
5	JBNU-CCLab	61.4%
6	DuluthNLP	53.3%
7	Stanford MLab	46.6%
8	niksss	44.2%
–	BERT (baseline)	45.7%

Table 5: Results for Task 1 (classification).

Rank	Team	Spearman's ρ
1	X-PuDu	0.807
2	PALI	0.785
3	HW-TSC	0.774
4	Nowruz	0.707
5	niksss	0.252
6	Stanford MLab	0.194

Table 6: Results for Task 2 (ranking).

Rank	Team	F1 (all)	F1 (w/o N)
1	X-PuDu	0.689	0.773
2	HW-TSC	0.661	0.749
3	PALI	0.654	0.749
4	Nowruz	0.624	0.714
5	JBNU-CCLab	0.551	0.627
6	DuluthNLP	0.533	0.608
7	Stanford MLab	0.466	0.514
8	niksss	0.442	0.494

Table 7: Classification results with/without NEUTRAL.

Rank	Team	Accuracy (#P≥2)
1	X-PuDu	75.2%
2	HW-TSC	73.2%
3	PALI	72.6%
4	Nowruz	71.6%
5	JBNU-CCLab	63.6%
6	DuluthNLP	62.6%
7	Stanford MLab	54.8%
8	niksss	60.0%

Table 8: Results for identifying contexts with multiple plausible fillers, based on individual model predictions.

also outperform our BERT-based baseline, which is a linear classification model based on the checkpoint provided by the Transformer library (Wolf et al., 2020) and fine-tuned on our training data.

6.1 Findings by Participants

In the following, we briefly summarize a couple of findings by task participants. More details can be found in the individual task description papers.

Different phenomena. The winning team, X-PuDu, found that different hyperparameters worked best depending on the phenomenon/extraction pattern. Based on this finding, different individual models were trained and combined in an ensemble.

Label distribution. Some teams, including DuluthNLP, noticed performance issues related to the distribution of labels in the development data. As a dedicated solution, DuluthNLP uses a decreased weight for the NEUTRAL label in the loss function.

NEUTRAL label. Team JBNU-CCLab reported that the NEUTRAL label is generally difficult to distinguish from other labels by different models. An underlying problem could be that the label represents instances that are seen as somewhat plausible by multiple annotators as well as instances that are seen as plausible by some annotators and implausible by others (see Section 4).

Noisy data. Team HW-TSC found that isolated training instances have the label NEUTRAL rather than PLAUSIBLE, even though the respective filler represents a human insertion (i.e., the filler can be found in the final version of the text in wikiHow). As the results of our human upper bound in Table 5 show, this is partly because the right label is sometimes not clear cut even for humans. We discuss this aspect in more detail in the next section.

6.2 Additional Evaluations

We perform two additional evaluations to assess the impact of the NEUTRAL label on system performance and to investigate the possibility of identifying whether multiple plausible clarifications exist by aggregating the predictions regarding individual clarifications.

Excluding NEUTRAL. For the evaluation without the NEUTRAL label, we calculate micro-averaged precision, recall and F_1-scores for the two labels PLAUSIBLE and IMPLAUSIBLE. The results in

Correct	#P≥2	Text	Fillers
8 (all)	✓	Galette des rois—or “King Cake” in English—is traditionally made to celebrate the __∅__ of Epiphany. Especially popular in France during the Christmas season, it is enjoyed elsewhere too.	✓ holidays ✓ Feast ? hours ? celebration ? proclamation
	✗	Let the __∅__ of shoes air dry. You can put them in front of a dehumidifier, a fan, or an open window, but avoid putting them in front of any type of heat source.	✓ pair ? pile ✗ shoes ✗ color ✗ end
0 (none)	✓	If you want a smoother surface, try a __∅__ of paper with a higher amount of grains, if you want a faster job but a rougher surface try a paper with a lower amount of grains.	✓ thickness ✓ piece ? fabric ? product ✗ pile
	✗	Your cucumber plant will also grow thin, light green shoots that help the plant grasp onto a surface and grow vertically. These __∅__ grow immediately next to the suckers.	✓ shoots ? fibers ? tendrils ? foliage ✗ bushes

Table 9: Examples of difficult and easy instances, selected based on how many systems classified them correctly.

terms of F_1-score are shown in Table 7. The results indicate that all systems perform substantially better in the evaluation setting that ignores NEUTRAL labels. The ranking is identical to the ranking in the evaluation including all labels. Considering only the PLAUSIBLE and IMPLAUSIBLE, Team X-PuDu achieves the highest micro-averaged F_1-score of 0.773. In the cases where their system predicts a non-NEUTRAL label, it is correct in 72.7% of cases (precision), and 82.5% of all non-NEUTRAL instances in the data received the correct prediction (recall).

Multiple clarifications. In our final evaluation, we examine whether system predictions can also be used to determine whether multiple plausible clarifications for a given context exist. For this, we consider the labels of each individual clarification and compare system outputs and annotations in terms of whether two or more clarifications for a cloze and its context received the label PLAUSIBLE. We show the result of this evaluation in terms of accuracy for each team in Table 8. Apart from the last two places, the teams rank in the same order as in the other evaluations. The best performing team, X-PuDu, correctly predicts whether two or more plausible clarifications exist for 75.2% of all cases. Table 9 shows examples that were correctly classified by all or none of the systems.

7 Conclusion

In this paper, we presented the task, data, participating systems, and results of the shared task on clarifying implicit and underspecified phrases in instructional texts. Our motivation for this task was to explore the possibility of testing different clarifications for plausibility. In particular, we were concerned with the question of whether two or more clarifications can be plausible and whether such cases can be detected automatically. To create a suitable dataset, we worked with and identified a set of revisions with manual clarifications, automatically generated possible alternatives, and then collected human plausibility ratings.

In total, 21 users participated in our shared task. We summarized the systems and results of 8 teams that submitted descriptions of their systems. The best systems from each group have in common that they are based on Transformer architectures or combine them in an ensemble. The best system achieved 68.9% accuracy, only 10.5 percentage points below a human upper bound. In additional evaluations, we have shown that an accuracy of up to 75.2% is achieved with respect to the detection of multiple plausible clarifications.

The results show that the presented task is a difficult one, but that many cases can already be modeled well by current state-of-the-art methods. There is further room for improvement with respect to both the data set and models: with respect to the data, it should be noted that the training set with less than 20k instances is relatively small and that there are many instances with a underspecified NEUTRAL label (36%). On the model side, we found that the participating teams make complementary contributions that may allow for additional improvements in combination.

One shortcoming of the task as presented and performed is that we only considered four forms of clarifications related to referring expressions. In addition, clarifications were assessed individually and judgements by different annotators were aggregated. In the long term, we believe that more forms of clarifications as well as individual differences regarding their plausibility need to be considered. Finally, future work will have to investigate under which circumstances multiple different clarifications are actually incompatible and can thus reveal potential sources of misunderstanding.

Acknowledgements

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

References

Tazin Afrin and Diane Litman. 2019. Identifying editor roles in argumentative writing from student revision histories. In *International Conference on Artificial Intelligence in Education*, pages 9–13. Springer.

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.

Talita Anthonio and Michael Roth. 2021. Resolving implicit references in instructional texts. In *Proceedings of the 2nd Workshop on Computational Approaches to Discourse*, pages 58–71, Punta Cana, Dominican Republic and Online. Association for Computational Linguistics.

Irshad Bhat, Talita Anthonio, and Michael Roth. 2020. Towards modeling revision requirements in wikiHow instructions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8407–8414, Online. Association for Computational Linguistics.

Amit Bronner and Christof Monz. 2012. User edits classification using document revision histories. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 356–366, Avignon, France. Association for Computational Linguistics.

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. In *Proceedings of ACL-08: HLT*, pages 789–797, Columbus, Ohio. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2019. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Johannes Daxenberger and Iryna Gurevych. 2012. A corpus-based study of edit categories in featured and non-featured Wikipedia articles. In *Proceedings of COLING 2012*, pages 711–726, Mumbai, India. The COLING 2012 Organizing Committee.

Alok Debnath and Michael Roth. 2021. A computational analysis of vagueness in revisions of instructional texts. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 30–35, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Charles Elkan. 2003. Using the triangle inequality to accelerate k-means. In *Proceedings of the 20th international conference on Machine Learning (ICML-03)*, pages 147–153.

Manaal Faruqui, Ellie Pavlick, Ian Tenney, and Dipanjan Das. 2018. WikiAtomicEdits: A multilingual corpus of Wikipedia edits for modeling language and discourse. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 305–315, Brussels, Belgium. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Francis Heylighen and Jean-Marc Dewaele. 2002. Variation in the contextuality of language: An empirical measure. *Foundations of science*, 7(3):293–340.

Omid Kashefi, Tazin Afrin, Meghan Dale, Christopher Olshefski, Amanda Godley, Diane Litman, and Rebecca Hwa. 2022. Argrewrite v. 2: an annotated argumentative revisions corpus. *Language Resources and Evaluation*, pages 1–35.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534, Berlin, Germany. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

James W. Pennebaker and Laura A. King. 1999. Linguistic styles: language use as an individual difference. *Journal of personality and social psychology*, 77(6):1296–1312.

Massimo Poesio, Jon Chamberlain, Silviu Paun, Juntao Yu, Alexandra Uma, and Udo Kruschwitz. 2019. A crowdsourced corpus of multiple judgments and disagreement on anaphoric interpretation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1778–1789, Minneapolis, Minnesota. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Merel Scholman and Vera Demberg. 2017. Crowdsourcing discourse interpretations: On the influence of context and the reliability of a connective insertion task. In *Proceedings of the 11th Linguistic Annotation Workshop*, pages 24–33, Valencia, Spain. Association for Computational Linguistics.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

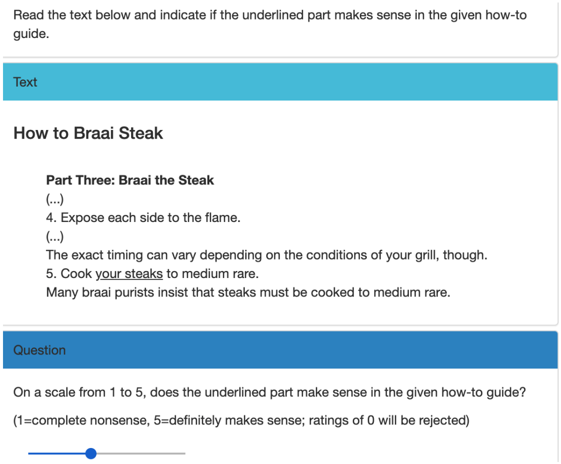
Diyi Yang, Aaron Halfaker, Robert Kraut, and Eduard Hovy. 2017. Identifying semantic edit intentions from revisions in Wikipedia. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2000–2010, Copenhagen, Denmark. Association for Computational Linguistics.

Fan Zhang and Diane Litman. 2015. Annotation and classification of argumentative writing revisions. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 133–143, Denver, Colorado. Association for Computational Linguistics.

Fan Zhang and Diane Litman. 2016. Using context to predict the purpose of argumentative writing revisions. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1424–1430, San Diego, California. Association for Computational Linguistics.

A Annotation Interface

The annotation interface for our crowdsourcing task is depicted in Figure 1. Annotators see and have to rate a single underlined clarification in its context.



Read the text below and indicate if the underlined part makes sense in the given how-to guide.

Text

How to Braai Steak

Part Three: Braai the Steak
(...)
4. Expose each side to the flame.
(...)
The exact timing can vary depending on the conditions of your grill, though.
5. Cook your steaks to medium rare.
Many braai purists insist that steaks must be cooked to medium rare.

Question

On a scale from 1 to 5, does the underlined part make sense in the given how-to guide?

(1=complete nonsense, 5=definitely makes sense; ratings of 0 will be rejected)

Figure 1: Interface for collecting annotations.

B CodaLab Leaderboard

In the main part of the paper, we only list results of participants who provided a description of their system(s) for the shared task. Table 10 shows a complete set of user names and results of the participants in the CodaLab competition, including users who did not submit a system description.

Team name	User name	acc.	ρ
X-PuDu	tt123	0.689	0.807
HW-TSC	Yinglu_Li	0.661	0.774
–	tiantaijian	0.661	0.763
–	fanxiaoxing	0.656	–
PALI	stce	0.654	0.785
–	hudou	0.641	–
–	huangwkk	0.631	0.774
Nowruz	mohammadmahdinoori	0.624	0.707
JBNU-CCLab	OrangeAvocado	0.614	–
–	CitizenTano	0.595	–
–	huawei_zhangmin	0.589	0.640
–	parkwonjae	0.554	–
–	lith	0.537	0.600
–	ywzhang_cr	0.537	0.600
DuluthNLP	Sakrah	0.533	–
Stanford MLab	patrickliu2011	0.466	0.194
–	Autism_PAFC	0.461	–
–	SelinaIW	0.456	–
niksss	niksss	0.442	0.252
–	andrei.manea	0.418	-0.109
–	tanigaki	0.395	0.415

Table 10: Oveview of results, including user submissions without a shared task system description.