How About Kind of Generating Hedges using End-to-End Neural Models?

Alafate Abulimiti[1,2], **Chloé Clavel**[3], **Justine Cassell**[1,4]

[1] INRIA, Paris [2] ENS/PSL <alafate.abulimiti@inria.fr>
[3] LTCI, Insitut Polytechnique de Paris, Telecom Paris <chloe.clavel@telecom-paris.fr>
[4] Carnegie Mellon University <justine@cs.cmu.edu>

Abstract

Hedging is a strategy for softening the impact of a statement in conversation. In reducing the strength of an expression, it may help to avoid embarrassment (more technically, "face threat") to one's listener. For this reason, it is often found in contexts of instruction, such as tutoring. In this work, we develop a model of hedge generation based on *i)* fine-tuning state-of-the-art language models trained on human-human tutoring data, followed by *ii)* reranking to select the candidate that best matches the expected hedging strategy within a candidate pool using a hedge classifier. We apply this method to a natural peer-tutoring corpus containing a significant number of disfluencies, repetitions, and repairs. The results show that generation in this noisy environment is feasible with reranking. By conducting an error analysis for both approaches, we reveal the challenges faced by systems attempting to accomplish both social and task-oriented goals in conversation.

1 Introduction

When people interact, they attend not just to the task at hand, but also to their relationship with their interlocutors (Tracy and Coupland, 1990). One key aspect of the relationship that people attend to, while engaging in contexts as diverse as sales (Gremler and Gwinner, 2008; Planken, 2005), education (Glazier, 2016; Murphy and Rodríguez-Manzanares, 2012) and healthcare (DiMatteo, 1979; Leach, 2005), is what is referred to as *rapport*, a sense of harmony and mutual understanding between participants in a conversation (Spencer-Oatey, 2005; Tickle-Degnen and Rosenthal, 1990). Indeed, higher levels of rapport are correlated with better performance in each of these domains. Zhao et al. (2014) describes rapport as built upon a base of mutual attentiveness, face management, and coordination. This base is built primarily by conversational strategies, or ways of speaking (including nonverbal and paraverbal behaviors) that manage rapport throughout a conversation. Key conversational strategies include self-disclosure, reference to shared experience, praise, and *hedging* — giving instructions or conveying information in an indirect manner when it might otherwise sound rude or overly demanding.

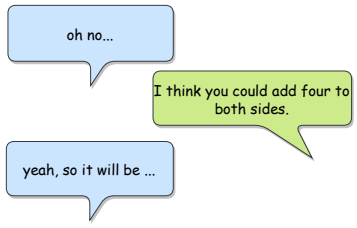

Figure 1: Hedging in peer tutoring

End-to-end large language models (LLM), of the kind that are increasingly popular and powerful, do a good job at carrying out the propositional or information-carrying aspects of conversation, and a relatively good job of maintaining the coherence of a conversation, but they are not as good at changing *how* they say something as a function of a relationship with the human user, while humans are, for the most part, quite good at this. However, since saying things in a specific manner - for example, through a hedge - helps task performance, it is an important topic for dialogue systems.

Linguists define hedges as a way of diminishing face threat (meaning the "positive social value a person effectively claims for himself" (Goffman, 1967) by attenuating the extent or impact of an expression (Brown and Levinson, 1987; Fraser, 2010). Figure 1 shows a typical example of hedging in a peer tutoring setting, where the tutor uses two hedges ("I think" and "could" rather than "should") to deliver a hint for the next step of solving an algebra equation.

Tutoring is one context in which hedges are found in abundance and where recognizing them might be important for intelligent tutoring systems, as attested by the number of computational ap-

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 877–892
July 9-14, 2023

proaches that attempt to do so (see section 2). Interestingly, even unskilled tutors use them. In fact, research on peer tutoring has shown that when rapport between a peer tutor and tutee is low, but the tutor is confident in his/her skills, that tutor tends to use more hedges, and this results in more problems attempted by the student and more problems successfully solved (Madaio et al., 2017).

In this paper, then, we work towards the development of a generation module for a virtual peer tutor that, like real peer tutors, is able to choose the manner of delivering information in such a way. Specifically, we address two research questions:

RQ1: How good are end-to-end large language models used alone for generating hedges when fine-tuned on a peer-tutoring dialogue dataset? Are the models able to implicitly learn when and how to generate hedges?

The first question may be answered by comparing the performance of various fine-tuned models. If the end-to-end models cannot learn to hedge implicitly, we might attempt to drive the models to generate the utterances by providing the correct labels. We assume that the correct labels can be provided by another module of the system, so we compare the reranking method with the fine-tuning method, as the former is simple, powerful, and widely used for text generation. Consequently, the second question is:

RQ2: Can we improve these models by using a reranking approach? If so, what are the remaining errors and why do they occur?

2 Related Work

Considerably more computational methods exist to determine *what* a dialogue system should say than *how* to say it. However, more recently, with the increased power of end-to-end models to find information and convey it accurately, we can now turn to ensuring that the end-to-end model simultaneously also meets social goals, to increase the impact and acceptability of what is conveyed.

2.1 Theoretical Approaches to hedges

As described above, a hedge can soften the impact of an utterance that might otherwise seem rude, such as a demand ("could you pass the salt") or an instruction ("you might want to pour the coffee over the sink"). Madaio et al. (2017) has attested to the frequent use of hedges in the peer-tutoring setting, and their positive impact on performance, perhaps because hedges in this context might reduce a tutee's embarrassment at not knowing the correct answer (Rowland, 2007).

In linguistic terms, hedging is a rhetorical strategy that attenuates the full force of an expression (Fraser, 2010) and for this reason, it has been covered in linguistic pragmatics and the study of politeness. Two main categories of hedges are identified in the literature: **Propositional Hedges** and **Relational Hedges** (Prince et al., 1982). Propositional Hedges (called **Approximators** by (Prince et al., 1982)) refer to uncertain (Vincze, 2014), fuzzy (Lakoff, 1975) and vague (Williamson, 2002) language use, such as "kind of". Relational Hedges (called **Shields** in (Prince et al., 1982)) indicate that the expression is subjective or an opinion, as in "*I think* that is incorrect". **Attribution Shields** are a subtype of relational hedges that attribute the opinion to others, such as "everyone says you should stop smoking". **Apologizers** (Raphalen et al., 2022) are apologies that mitigate the strength of an utterance, as in "I'm sorry but you have to do your homework".

While the different types of hedges operate in different ways, they all serve the same mitigation functions in conversation. For this reason, in what follows — a first attempt at generating hedges — we collapse the different sub-classes and refer only to hedges and non-hedges.

2.2 Computational Approaches

Some prior work has looked at the detection of conversational strategies and in particular work by Zhao and colleagues (Zhao et al., 2014, 2016b,a). Madaio et al. (2017) built a classifier to detect hedging and achieved an accuracy of 81%. Recent work by Raphalen et al. (2022) improved the detection of different types of hedges and achieved a weighted F1 score of 0.97.

Hedging is a particular kind of indirectness, and therefore as we look at prior work in the area, we include approaches to the generation of indirect speech. The plan-based generation of indirect speech acts has existed almost as long as dialogue systems themselves (Clark, 1979; Brown, 1980; Perrault, 1980). More recently, other relevant aspects of politeness have also been addressed. For example, Porayska-Pomsta and Mellish (2004) operationalized the important notion of face in politeness theory to generate polite sentences with a template pool. Although contemporary dialogue

systems tend to integrate indirect speech (Miehle et al., 2022; Briggs et al., 2017), generating hedges with powerful language models, and particularly as a function of the social context, has not been explored. Our desire to look at the social context leads us to train on spontaneous dialogue that is substantially noisier, owing to natural conversational phenomena such as disfluency. This differs from the majority of prior work, trained on written or acted corpora (Li et al., 2017; Rashkin et al., 2019).

2.3 Generation Techniques

Different techniques have been used in the past to generate responses of a particular kind for dialogue systems. Madaan et al. (2020) used n-gram TF-IDF to identify source style words and generate target politeness style utterances by replacing these words. Niu and Bansal (2018) generated politeness formulations by using reinforcement learning with a trained politeness classifier. Similar to our approach, the explicit knowledge of politeness is only given to the classifier. Liu et al. (2021) constructed an emotional support dataset with eight different dialogue strategies and fine-tuned the pre-trained language models by connecting the label tokens to the beginning of each utterance in order to create a dialogue generator that can produce the target responses without focusing on the social context.

The reranking method is also widely used in text generation tasks. Hossain et al. (2020) used a simple and effective pipeline where they retrieved the original texts from the database, then edited with a Transformer (Vaswani et al., 2017) model, and then reranked the text by generation scores. Soni et al. (2021) first applied reranking to conversational strategy generation by controlling the level of self-disclosure in the outputs of DialoGPT (Zhang et al., 2020b). The authors of LaMDA (Thoppilan et al., 2022) used various classifiers to rerank and filter out inappropriate responses. Recently, ChatGPT (OpenAI, 2022) used reinforcement learning with human feedback, and has shown impressive performance.

In the articles above, most algorithms were trained on written dialogue datasets, which facilitated the task. However, our spontaneous dialogue dataset may lead the way for cutting-edge models trained on a real-world, face-to-face interactional dataset.

3 Methodology

3.1 Task Description

Let $D = \{d_1, d_2, d_3, ...d_n\}$ be a set of dialogues, where each dialogue $d = \{u_1, u_2, u_3...u_m\}$ is composed of m turns, where u_i is a turn. Each tutor turn (and each tutee turn, although we will not examine the tutee turns further here) is labeled as hedge or non-hedge; we call l_i the label of u_i. A fixed window size ω of the dialogue history is assigned to each utterance: $h_i = \{u_{max(1,i-\omega)}, u_{i-\omega+1}, ...u_{i-1}\}$. The goal of this work is to train a generator (G) that can produce a tutor's utterance u_i' that matches a given hedge strategy (i.e., hedge or non-hedge) l_i, according to the dialogue history h_i.

3.2 Corpus

The dataset we used in the current work is the same as that used in our prior work (Raphalen et al., 2022; Goel et al., 2019; Zhao et al., 2014). 24 American teenagers aged 12 to 15, half boys and half girls, were assigned to same-gender pairs. They took turns tutoring each other in linear algebra once a week for five weeks, for a total of 60 hours of face-to-face interaction. Each interaction was composed of two tutoring periods, where the teens took turns being the tutor, with a social period at the beginning and between the two tutoring periods. For the purposes of the earlier work the corpus was annotated for hedges, as well as the subcategories of hedges, at the clause level. For our purposes, since generation happens at the level of the turn, we merge the clauses and their labels into speaker turns and turn-level hedge labels (see Appendix A for the merge strategy).

Our goal is to create a hedge generation module that can produce an appropriate hedge strategy for a tutor giving an instruction, according to what has been said before as indicated by the dialogue history. For this reason we kept all turns in the dialogue history, even though our model is trained to generate only the tutor's turns (and not those of the tutee). There are 6562 turns in these interactions, of which 5626 contain non-hedges and 936 hedges.

Being authentic interaction, there are disfluencies ("so just yeah just um"), repetitions ("that would be then that would be"), repairs ("oh wait, actually the x would go here"), and other spoken phenomena such as one-word clauses. These phenomena make generating hedges challenging since the language models we use are primarily trained

on written dialogues, which do not contain most of these features. However, our work allows us to see how far we can go with authentic spoken data.

3.3 Methods

We combine two techniques for generating the tutor's turn: *Fine-tuning* an existing generation model and *Re-ranking* the generated outputs to match the desired hedge strategy.

3.3.1 Fine Tuning Method

First, we want to evaluate how well the model performs when hedge information is implicitly taught through fine-tuning. We fine-tuned the generation model with the training set of the peer-tutoring corpus. Each utterance $u_i = (w_1, ..., w_n)$ is composed of n tokens, the dialogue history h_i as input to the generation model. We apply cross-entropy loss between u_i and u'_i, where $u' \in R^{|V|}$, V is the vocabulary.

$$J(u_i, u'_i) = -\frac{1}{n} \sum_{j=1}^{j=|V|} u_{i,j} \log(u'_{i,j}) \qquad (1)$$

3.3.2 Reranking Method

Since a hedge classifier was developed for prior work in our lab (Goel et al., 2019; Raphalen et al., 2022), we can use it to determine whether a generated text is a hedge or not and then inform the generator of the decision in order to regulate the output. This is known as reranking, and is what we use here as our second generation strategy.

1) We first pretrain our generator as in fine tuning. We then apply this generator to the test set to generate 50[1] candidate utterances for each dialogue history (Figure 2). 2) These candidates are first ranked by their sentence scores (i.e., the final outputted token's log probability for each sentence). 3) We then use the hedge classifier described above to filter out the utterances that do not match the selected strategy (i.e., hedge or non-hedge). 4) We keep utterances that match the selected hedge strategy. If more than one candidate matches the strategy, we pick the first one that matches, which means the one with the highest sentence score. 5) If none of the candidates matches the selected hedge strategy, we output the one that has the highest sentence score.

[1] See Appendix C for the details

4 Experimental Setting

4.1 Data Processing

We randomly split the final dataset based on a 60:20:20 ratio. Of these, 60% is the training set, 20% is the validation set, and 20% is the test set.

Since our dataset is highly unbalanced, if we used it as is the results would be too biased towards non-hedges. In that approach the gap between the results of different models would not be clear because non-hedges are so much more frequent. For this reason, we manually balance by randomly selecting 235 non-hedge turns to balance the 235 hedges in the test set, and combine the data to form a new balanced test set. On the other hand, in order to have a large enough training set, we retain all tutor turns from the complete dataset, which therefore consists of 701 hedge turns and 4455 non-hedge turns, resulting in a dataset that is very skewed, but has more turns.

While the complete dataset contains a relatively small number of hedge turns, we believe that preserving the natural data distribution is crucial for addressing our first research question. Underscoring the wisdom of this approach, the results we obtained on perplexity and the BARTscore (that are indicative of fluency in the generated responses, as described below) demonstrate that the models were able to generate responses with reasonable fluency and quality despite the small number of hedge turns.

4.2 SOTA Pretrained Language Models

We compare the performance of different state-of-the-art (SOTA) free open-source pretrained models as our generators: BART, DialoGPT, and BlenderBot. BART (Lewis et al., 2020) uses an encoder-decoder architecture, trained on books and Wikipedia data, and performs well on tasks as varied as Q&A (SQuAD (Rajpurkar et al., 2016)), text generation, text classification (MNLI (Williams et al., 2018)), and text summarization tasks (ELI5 (Fan et al., 2019)). It is pretrained by distorting the format of the input text in various ways, and this training helps us to visualize its possible application to noisy spontaneous spoken dialogues. DialoGPT (Zhang et al., 2020b) is a dialogue version of GPT-2 (Radford et al., 2019), an autoregressive language model with a multi-layer Transformer (Vaswani et al., 2017) decoder as its model architecture. It is trained on 140 million conversational exchanges extracted from Reddit comment

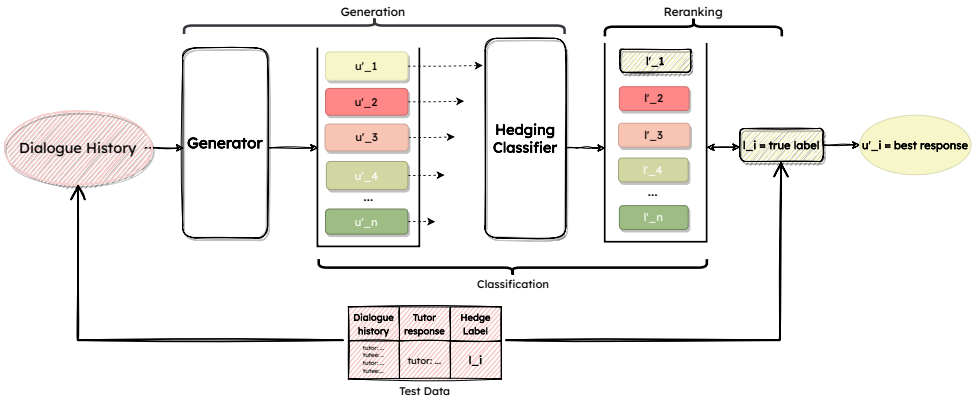

Figure 2: Reranking method

threads. BlenderBot (Roller et al., 2021) uses the standard Seq2Seq Transformer architecture, but incorporates a number of dialogue training sets: Empathetic Dialogue (Rashkin et al., 2019), PersonaChat (Zhang et al., 2018), ConvAI2 (Dinan et al., 2020), and other datasets that, while largely handcrafted, focus on personality and emotions, enabling it to potentially develop some version of social skills.

4.3 Evaluation Metrics

To evaluate performance, we used the most widely used set of reference-based metrics for natural language generation tasks (Liu et al., 2021; Ziems et al., 2022). Since these metrics have not been used for conversational strategies, we add an unsupervised reference-free metric, the BART score (Yuan et al., 2021). The BART score formulates the evaluation process as a text generation task using a pre-trained model. The score represents the probability of generating a hypothesis given a source text. The higher BART score represents better text from different perspectives (e.g., informativeness, factuality). In this paper, we denote the dialogue history as the source text and the generated utterance as the hypothesis. For comparison, we calculate the BART score between the dialogue history and the real response in the test dataset, giving a result of -6.44. We also evaluated the relevance of the generated hedge strategy using an F1 score. The results using these metrics are presented in Table 2. The detailed description of the metrics used is in Appendix B.

4.4 Human Evaluation

While the metrics described above are important for comparison with the performance of other work in the field, they do not obviate the need for human annotation. We therefore asked two annotators to ignore sub-categories and annotate only hedge or non-hedge on each tutor turn of the model's output, with access to 4 prior turns of the dialogue history. During a training phase the annotators reached an inter-rater reliability of over .7 Kripendoff's alpha (Krippendorff, 2004) which indicates substantial agreement. One of the annotators then finished the remainder of the annotation. We computed the F1 scores for the label of the generated utterances with respect to the real tutor turn's label. A higher F1 score indicates that the approach is better suited to generate the correct hedge strategy (see Table 2). We also asked the annotators to pay attention to whether the output was unnatural and to note it if so. The annotators reported no concerns with the naturalness of the generated utterances.

The concept of fluency has recently gained popularity in the dialogue community (Li et al., 2019; See et al., 2019), but the current definition of fluency varies. More fundamentally, evaluations of this kind are more applicable to written text or scripted dialogues (Pang et al., 2020; D'Haro et al., 2019). as they cannot handle disfluencies (e.g., hesitations, repetitions, false starts) of the kind that are common in spontaneous spoken dialogues, and that may serve to give the speaker time to plan the next utterance (Biber et al., 1999; Thornbury and Slade, 2006). We therefore did not assess fluency in this work.

5 Results

5.1 RQ1: How well do end-to-end models perform alone for generating hedges?

Table 2 compares the performance of the generation models. BlenderBot outperforms the other 2 models on most metrics,although with similar per-

formance to DialoGPT, on BLEU and ROUGE-L. The discrepancy between BlenderBot and BART in each score is relatively wide. This discrepancy is most apparent on measures that compute scores based on n-gram-level overlaps (BLEU, ROUGE). To find the reason for this discrepancy, we calculate the average length of the outputs of the 3 models and observe 5.2 words for BART, 11.8 words for BlenderBot, and 14.5 words for DialoGPT, while the average length of the tutor's utterances in test data is 15.2 words. The average length of the output of DialoGPT is therefore close to that of the test set. This further explains DialoGPT's strong performance on the BLEU and ROUGE scores. On the other hand, BART tends to generate shorter turns, consequently demonstrating lower scores on metrics that require the calculation of repetition grams to yield scores. Note that in similar tasks, the best model was Blenderbot with a BLEU 2 score of 6.21, in the case of emotional support conversational strategy generation (Liu et al., 2021), while DialoGPT reached 5.52. The best score in the positive text reframing task, meanwhile, was 11.0 for BLEU 1 (Ziems et al., 2022), while BART reached 10.1 and GPT-2 reached 4.2.

Table 1 shows that BART has the lowest perplexity score, indicating that BART is more adaptive to our dataset compared to the other two models. This may be due to its pre-training approaches (see Section 4.2) that corrupt input texts with an arbitrary noising function. These approaches enable more accurate predictions in our noisy real-world dataset.

BART	BlenderBot	DialoGPT
34.9	69.3	72.4

Table 1: Language Model (LM) Perplexity (the lower is the better

In response to our first research question, then, the performance of all three models was comparable but very limited. This suggests that the fine-tuning approach does not allow language models to learn hedge knowledge implicitly.

We therefore next turn to an approach that may improve performance by screening utterances with a given label.

5.2 RQ2: Does reranking improve hedge generation?

Table 2 shows the performance of each model for the reranking method. BlenderBot once again performs well on all metrics and has a virtually identical F1 score to BART. Additionally, we find some interesting similarities among models: 1) BlenderBot and DialoGPT outperform BART in both the fine-tuning and the reranking methods (Table 2) with respect to reference-based metrics such as BLEU, ROUGE-L, etc., and 2) DialoGPT still underperforms the other two models in terms of F1 score, and in the reranking condition the gap widens.

Models / Metrics	BlenderBot	DialoGPT	BART	R_BlenderBot	R_DialoGPT	R_BART
BLEU_1	11.2	**11.4**	2.7	**12.3**	10.9*	6.0*
BLEU_2	**5.8**	4.7	1.5	**6.2**	3.9*	3.1*
ROUGE-L	8.6	**9.1**	8.1	**11.0**	8.4	9.7
CHRF	**17.6**	17.0	9.3	**17.6***	17.5*	12.2*
BARTScore	**-3.92**	-5.62	-4.33	**-3.98***	-4.79	-4.24
BERTScore	**39.9**	38.3	38.5	**40.5**	37.5	39.4
F1Score (human evaluation)	**0.54**	0.41	0.44	0.84	0.64	**0.85**

Table 2: Results of the fine-tuned models and reranking method applied to the fine-tuned models. *means this result is significantly different from the fine-tuning method ($p < .05$)

This result could suggest that 1) the pretraining of the models (i.e., DialoGPT, BlenderBot) on dialogue datasets may help to generate longer utterances, and therefore to improve the reference-based metrics performance, and 2) the autoregressive model (e.g., DialoGPT) may not be suitable for the generation of social dialogue such as hedges.

5.3 Comparing Fine-tuning and Reranking

To summarize results on the fine-tuning versus reranking approaches we observe that: 1) With the help of a hedge classifier, the reranking approach can do a good job at generating hedges, 2) BlenderBot is better suited to the task of generating long utterances, as described in Section 5.1. This could be because BlenderBot is pretrained with various social dialogue datasets, giving it a certain ability to generate the social aspects of dialogue.

Table 2 shows that models deployed with the reranking method have relatively higher or comparable Bart scores, but greatly improved performance on the F1 score (from .54 to .85). This result, too, underscores the advantages of the reranking method.

5.4 Error Analysis

While BlenderBot showed strong performance when using reranking, a certain number of generated utterances still did not match the real tutor

labels. When a matching utterance type cannot be found in a limited pool of candidates, we could have chosen to increase the candidate pool to promote the probability of selecting a match. However, in this early effort to generate hedges, we want to ensure sufficient quality in the generated output but also explore the limitations of current language models for generating socially relevant phenomena on the basis of a spontaneous spoken interaction dataset.

We can learn about the limitations of these models by examining places where the system did not generate the desired strategy (that is, generated a hedge when the real tutor did not or vice versa). We first divide these strategy mismatches into *over-generation errors*, where the generator generates a hedge where it should not and *under-generation errors* when it does not generate a hedge but should. Among the 1395 annotated turns outputted by the 3 generators, there are 13.3% of *over-generation errors* and 86.7% *under-generation errors*. These errors are particularly interesting in the context of reranking, as it relied strongly on the hedge classifier. The hedge classifier selected the most suitable utterances, and yet the model still produced the wrong strategy - or at the very least mismatches with the strategy of the real tutor.

Therefore, we analyze the generated utterances corresponding to these two types of errors and identify two potential causes.

First, there are still some places where the model generates a hedge where it should generate a non-hedge. As we mentioned in Section 4.4, we invited humans to annotate the models' outputs in terms of hedge labels. We compare the human-annotations of the model output (where they labeled the output as hedge or non-hedge) with the output of the BERT-based classifier on the same generated utterances to calculate the F1 score. We find that there is a difference of about 9 points between the F1 score for human annotation (85%) shown in Table 2, and the F1 score for the same BERT-based hedge classifier (94%) reported in Raphalen et al. (2022). We assume that the classifier we used may have misclassified some generated utterances and we therefore label them as **Classification Errors**. This category accounts for 92.5% of *over-generation errors*, and 15.3% of *under-generation errors*.

Second, the basic functionality of an end-to-end language model of this kind is to produce the most coherent next utterance based on the dialogue history. This may result in the language model privileging coherence of content over style of delivery. That is, the model may not be able to find an appropriate strategy match among the coherent candidates, even when the candidate pool size is 50. We label this a **Goal Mismatch** as the propositional or content coherence goals of the system may be trumping the social goals, We found 84.7% in *under-generation errors* and 7.5% in *over-generation errors*. 18% of the cases where the pool did not include the right strategy.

An example of each type of error is given in Figure 3. The first example belongs to the **Classification Error** type, where the classifier misclassified the system response (i.e. "We just found that the answer is two x equals three") as a hedge. In the second example, the tutor is trying to help the tutee to approach the answer step by step, but the tutee cannot come up with a worked idea. Here it is clear that the tutee is flailing and it is therefore probably not advisable to increase the student's stress with a volley of questions that the tutee can clearly not answer. The tutor thus uses a hedge as a response. Conversely, the generator produces a question. The generated utterance is "What do you think we should do, what's the next step". This example corresponds to our **Goal Mismatch Error**. It shows that the generator may not understand the social context, but is looking for a coherent response.

The **Goal Mismatch Error** is perhaps the most interesting of the errors, and thus to verify our hypothesis — that the coherence goals of the models may impede the social goals — we looked into the nature of the relationship between rapport (between tutor and tutee) and the generation of hedges. As described above, Madaio et al. (2017) found that hedges are generated when rapport is low. Since our corpus contained rapport annotations for every 30 seconds of the interaction, we looked at the rapport level in play when the model over-generated and under-generated hedges. Since rapport is annotated from 1 to 7 in the dataset, for convenience, we divided it into 3 levels: high (5-7), medium (3-5), and low rapport (1-3), as shown in Table 3.

Type \ Rapport	**High**	**Medium**	**Low**
Over-generation	0	3	0
Under-generation	13	130	75

Table 3: Goal Mismatch Errors Distribution

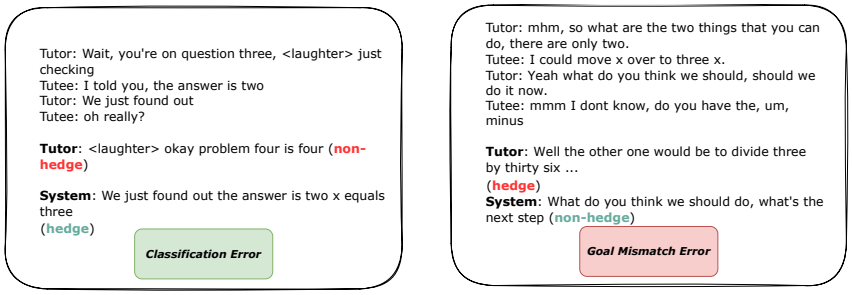

Figure 3: Strategy Mismatch Errors for Reranking Method

As only 3 errors appear in the category of *over-generation error*, we cannot obtain a meaningful conclusion due to size. However, the generators generate fewer hedges when rapport is low, an *under-generation error*, in contradiction to studies showing that speakers are more careful about threatening the face of (or embarrassing) their interlocutors when the social bond between them is weak (Madaio et al., 2017). We believe that this is because more hedges are found in low rapport interaction. Therefore, we count the hedge distribution of the low and high rapport interaction in the test dataset. 264 hedges are found in low rapport interaction, and 42 in high rapport interaction. This distribution corresponds to the fact that a hedge is most likely to happen in low rapport interactions. The under-generation errors are the cases where there should be hedges but non-hedges were generated. In the test dataset, more hedges occur in low rapport, and the generator under-generates more in low rapport, because there are more hedges that should be generated in low rapport. So, the generators make more errors in low rapport interaction due to an imbalance in hedge distribution between low and high rapport interaction.

Goal Mismatch error directly addresses our primary question 1: How effectively do end-to-end models perform when generating hedges on their own? Due to this fundamental discrepancy between competing goals, end-to-end language models are unable to inherently learn and discern when to apply hedges appropriately.

5.4.1 Lexical Diversity of the Generated Output

As we have seen, LLMs can generate a hedge or non-hedge with the help of the reranking method. However, do language models spontaneously use different types of hedges in a human-like way? To investigate this question, we applied the rule-based hedge classifier from (Raphalen et al., 2022) to automatically annotate the utterances generated by models in subcategories of hedges (as defined in Section 2.1), and we compare the models' and humans' distributions of different hedge strategies. The rule-based classifier used linguistic patterns to identify each hedge subcategory. We have preferred here to use the rule-based classifier rather than the machine learning classifiers to avoid the dependence on and bias of probabilistic learning-based classifiers. Indeed, learning-based classifiers may be biased towards predicting the categories that are the most frequent in the dataset. Furthermore, the rule-based classifier reaches a 94.7 F1 score (Raphalen et al., 2022), which is comparable to the best performance (96.7 F1 score) using the Light Gradient-Boosting Machine (LGBM) (Ke et al., 2017) classifier.

The above results show that the model can spontaneously learn to use different types of hedges. Indeed, the models are capable of carrying out linguistic diversity on hedges based on learning from real human dialogues.

6 Conclusion and Future Work

In this paper, we have shown that the reranking method helps LLMs to generate hedges — an important social conversational strategy that can avoid face threats towards an interlocutor by attenuating an impact of an expression. We find that an implicit fine-tuning approach (i.e., without any supervision by a hedge label) is not sufficient for generating hedges, but a reranking method significantly improves performance in generating hedges, with a final F1 score of .85 for the BART model and .84 for the BlenderBot model. We also performed an error analysis on the generated results and found that two types of errors occur in the reranking method: **Classification**, and **Goal Mismatch**. The vast majority of errors fall into the category of Goal Mismatch, indicating an important conflict between

contemporary language models' primary goal of ensuring coherence and the social goal of managing face, which is indispensable for human conversation. While we were able to generate hedges, we were not able to necessarily generate them where they were needed most. That is, conversational strategies are adaptive in the sense that they respond to conversational strategies uttered by the previous speaker (Zhao et al., 2014). We conclude that, going forward, we will need a way of adding an underlying representation of the social state of the dialogue to improve dialogue generation.

In this paper we addressed the question of how to generate hedges, but when to generate hedges remains an important and unexplored question. In future work, we may first explore the temporal relationships between the hedge and other conversational information (e.g., other conversational strategies, level of rapport) by sequential rule mining techniques, then apply RL-based methods to investigate in a more detailed manner the optimal way to predict where hedges should occur. In this context, we note that ChatGPT can generate a hedge when requested explicitly to do so, but does not generate hedges of its own volition (so to speak), for example, when face-threatening acts such as instruction are engaged in.

We began this paper by describing the need for hedges in instructional dialogues such as those engaged in by intelligent tutoring systems. The current dataset consists of authentic real-world tutoring sessions, but as carried out by untrained teenagers. We note that peer tutoring is a powerful method of teaching, used in classrooms around the world, and previous work shows that when untrained peer tutors use hedges, their tutees attempt more problems and solve more problems correctly (Madaio et al., 2017). However, they are inexperienced and so in future work it will be important to investigate the interaction between trained tutors and tutee as well, for instance, by using the Teacher-Student Chatroom Corpus (Caines et al., 2020). We believe that the methods and results from the current work will facilitate the investigation of expert tutors in future research.

Broader Impact

Since the 1990s, research has shown the the importance of intelligent tutoring systems as effective learning environment,s and supports for classroom learning (Anderson et al., 1995). Peer tutoring plays a powerful role as well, as peer tutors can motivate learners to try harder, as well as helping them to succeed, and it is particularly effective for low-achieving learners (Cassell, 2022). But virtual peer tutors have not yet achieved their potential, in part because of the difficulty of generating the social infrastructure of peer learning as well as the content of the matter being tutored. This paper, whose data comes from a corpus of peer tutoring dialogues, should therefore be seen as a step in the right direction.

Acknowledgments

We thank the anonymous reviewers for their helpful feedback. We express sincere gratitude to the members of the ArticuLab at Inria Paris for their invaluable assistance in the successful completion of this research, and to the members of the ArticuLab at Carnegie Mellon Pittsburgh for answering our questions about their prior work. This study received support from the French government, administered by the Agence Nationale de la Recherche, as part of the "Investissements d'avenir" program, with reference to ANR-19-P3IA-0001 (PRAIRIE 3IA Institute).

Limitations

Several limitations apply to the current study. While research shows that multimodal signals play an important role in conversational strategies (Zhao et al., 2016b), we did not take them into account. It is an open question as to how to render large language models capable of generating multimodal behaviors. A second limitation concerns the recent arrival on the scence of ChatGPT, that has shown impressive performance. However the models are not free, and therefore were not included. As noted above, another important limitation is the untrained status of the tutors in our corpus, who are teenagers, and not trained tutors. Their use of hedges, therefore, comes from their knowledge of everyday social interaction, and not from expertise in teaching. In looking at the data, we find a few places where, as instructors ourselves, we believe that a hedge is important, even though the real (teenage) tutor did not use one.

The last limitation is that, while we focused only on generating hedge or non-hedge, there are actually 3 different kinds of hedges, that function differently. We hope to extend this work and take

advantage of a text style transfer technique to generate more kinds of hedges in future work.

Ethical Statement

The corpus used here comes from earlier work by the last author and her colleagues, and was used in accordance with the original experimenters' Institutional Review Board (IRB). Those experimenters also anonymised the data, removing any identifying information. A pixelated example of the video data is available at github.com/neuromaancer/hedge_generation. To counteract potential gender bias concerning the use of hedges in peer tutoring, the data was collected from equal number of boys and girls. In text generation tasks, it is important to be aware of the potential risk of generating inappropriate content. We believe that, in fact, hedges used by tutors are perhaps the least likely conversational strategy to be inappropriate, as they are the most polite and "delicate" conversational moves. But, more generally, considerable additional work would be needed to filter out all inappropriate language for safe tutoring systems that engage in social and task interaction.

References

John R Anderson, Albert T Corbett, Kenneth R Koedinger, and Ray Pelletier. 1995. Cognitive tutors: Lessons learned. *The journal of the learning sciences*, 4(2):167–207.

Chris Berry and Allen Brizee. 2010. Identifying independent and dependent clauses. *Purdue OWL*.

Douglas Biber, Stig Johansson, Geoffrey Leech, Susan Conrad, Edward Finegan, and Randolph Quirk. 1999. *Longman grammar of spoken and written English*, volume 2. Longman London.

Gordon Briggs, Tom Williams, and Matthias Scheutz. 2017. Enabling robots to understand indirect speech acts in task-based interactions. *Journal of Human-Robot Interaction*, 6(1):64–94.

Gretchen P Brown. 1980. Characterizing indirect speech acts. *American Journal of Computational Linguistics*, 6(3-4):150–166.

Penelope Brown and Stephen C. Levinson. 1987. *Politeness: Some universals in language usage, volume 4*. Cambridge university press.

Andrew Caines, Helen Yannakoudakis, Helena Edmondson, Helen Allen, Pascual Pérez-Paredes, Bill Byrne, and Paula Buttery. 2020. The teacher-student chatroom corpus. In *Proceedings of the 9th Workshop on NLP for Computer Assisted Language Learning*, pages 10–20.

Justine Cassell. 2022. Socially interactive agents as peers. In *The Handbook on Socially Interactive Agents: 20 years of Research on Embodied Conversational Agents, Intelligent Virtual Agents, and Social Robotics Volume 2: Interactivity, Platforms, Application*, pages 331–366.

Stanley F Chen, Douglas Beeferman, and Roni Rosenfeld. 1998. Evaluation metrics for language models.

Herbert H Clark. 1979. Responding to indirect speech acts. *Cognitive psychology*, 11(4):430–477.

Luis Fernando D'Haro, Rafael E Banchs, Chiori Hori, and Haizhou Li. 2019. Automatic evaluation of end-to-end dialog systems with adequacy-fluency metrics. *Computer Speech & Language*, 55:200–215.

M Robin DiMatteo. 1979. A social-psychological analysis of physician-patient rapport: toward a science of the art of medicine. *Journal of Social Issues*, 35(1):12–33.

Emily Dinan, Varvara Logacheva, Valentin Malykh, Alexander Miller, Kurt Shuster, Jack Urbanek, Douwe Kiela, Arthur Szlam, Iulian Serban, Ryan Lowe, et al. 2020. The second conversational intelligence challenge (convai2). In *The NeurIPS'18 Competition: From Machine Learning to Intelligent Conversations*, pages 187–208. Springer.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Bruce Fraser. 2010. Pragmatic competence: The case of hedging. new approaches to hedging.

Rebecca A Glazier. 2016. Building rapport to improve retention and success in online classes. *Journal of Political Science Education*, 12(4):437–456.

Pranav Goel, Yoichi Matsuyama, Michael Madaio, and Justine Cassell. 2019. i think it might help if we multiply, and not add. In *Detecting indirectness in conversation. In 9th International Workshop on Spoken Dialogue System Technology*, page 27–40. Springer.

Erving Goffman. 1967. *Interaction Ritual*, chapter On Face-Work. Pantheon, New York.

Dwayne D Gremler and Kevin P Gwinner. 2008. Rapport-building behaviors used by retail employees. *Journal of Retailing*, 84(3):308–324.

Nabil Hossain, Marjan Ghazvininejad, and Luke Zettlemoyer. 2020. Simple and effective retrieve-edit-rerank text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2532–2538, Online. Association for Computational Linguistics.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30.

Klaus Krippendorff. 2004. Reliability in content analysis: Some common misconceptions and recommendations. *Human communication research*, 30(3):411–433.

George Lakoff. 1975. Hedges: A study in meaning criteria and the logic of fuzzy concepts. In *Contemporary research in philosophical logic and linguistic semantics*, pages 221–271. Springer.

Matthew J Leach. 2005. Rapport: A key to treatment success. *Complementary therapies in clinical practice*, 11(4):262–265.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. DailyDialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Zekang Li, Cheng Niu, Fandong Meng, Yang Feng, Qian Li, and Jie Zhou. 2019. Incremental transformer with deliberation decoder for document grounded conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 12–21.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Siyang Liu, Chujie Zheng, Orianna Demasi, Sahand Sabour, Yu Li, Zhou Yu, Yong Jiang, and Minlie Huang. 2021. Towards emotional support dialog systems. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3469–3483, Online. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Aman Madaan, Amrith Setlur, Tanmay Parekh, Barnabas Poczos, Graham Neubig, Yiming Yang, Ruslan Salakhutdinov, Alan W Black, and Shrimai Prabhumoye. 2020. Politeness transfer: A tag and generate approach. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1869–1881, Online. Association for Computational Linguistics.

Michael Madaio, Justine Cassell, and Amy Ogan. 2017. The impact of peer tutors' use of indirect feedback and instructions. Philadelphia, PA: International Society of the Learning Sciences.

Juliana Miehle, Wolfgang Minker, and Stefan Ultes. 2022. When to say what and how: Adapting the elaborateness and indirectness of spoken dialogue systems. *Dialogue & Discourse*, 13(1):1–40.

Elizabeth Murphy and María A Rodríguez-Manzanares. 2012. Rapport in distance education. *International Review of Research in Open and Distributed Learning*, 13(1):167–190.

Tong Niu and Mohit Bansal. 2018. Polite dialogue generation without parallel data. *Transactions of the Association for Computational Linguistics*, 6:373–389.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue.

Bo Pang, Erik Nijkamp, Wenjuan Han, Linqi Zhou, Yixian Liu, and Kewei Tu. 2020. Towards holistic and automatic evaluation of open-domain dialogue generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3619–3629.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

C Raymond Perrault. 1980. A plan-based analysis of indirect speech act. *American Journal of Computational Linguistics*, 6(3-4):167–182.

Brigitte Planken. 2005. Managing rapport in lingua franca sales negotiations: A comparison of professional and aspiring negotiators. *English for Specific Purposes*, 24(4):381–400.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Kaśka Porayska-Pomsta and Chris Mellish. 2004. Modelling politeness in natural language generation. In *International Conference on Natural Language Generation*, pages 141–150. Springer.

Ellen F. Prince, Joel Frader, and Charles Bosk. 1982. On hedging in physician-physician discourse. *Linguistics and the Professions*, 8(1):83–97.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Yann Raphalen, Chloé Clavel, and Justine Cassell. 2022. "You might think about slightly revising the title": Identifying hedges in peer-tutoring interactions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2160–2174, Dublin, Ireland. Association for Computational Linguistics.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325, Online. Association for Computational Linguistics.

Tim Rowland. 2007. well maybe not exactly, but it's around fifty basically? In *Vague language in mathematics classrooms. In Vague language explored*, page 79–96. Springer.

Abigail See, Stephen Roller, Douwe Kiela, and Jason Weston. 2019. What makes a good conversation? how controllable attributes affect human judgments. 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 1702–1723.

Mayank Soni, Benjamin Cowan, and Vincent Wade. 2021. Enhancing self-disclosure in neural dialog models by candidate re-ranking. *ArXiv preprint*, abs/2109.05090.

Helen Spencer-Oatey. 2005. (im)politeness, face and perceptions of rapport: Unpackaging their bases and interrelationships. 1(1):95–119.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *ArXiv preprint*, abs/2201.08239.

Scott Thornbury and Diana Slade. 2006. *Conversation: From description to pedagogy*. Cambridge University Press.

Linda Tickle-Degnen and Robert Rosenthal. 1990. The nature of rapport and its nonverbal correlates. *Psychological inquiry*, 1(4):285–293.

Karen Tracy and Nikolas Coupland. 1990. Multiple goals in discourse: An overview of issues. *Journal of Language and Social Psychology*, 9(1-2):1–13.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Veronika Vincze. 2014. Uncertainty detection in natural language texts. *PhD, University of Szeged*, 141.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Timothy Williamson. 2002. *Vagueness*. Routledge.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020a. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020b. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics.

Ran Zhao, Alexandros Papangelis, and Justine Cassell. 2014. Towards a dyadic computational model of rapport management for human-virtual agent interaction. In *International conference on intelligent virtual agents*, pages 514–527. Springer.

Ran Zhao, Tanmay Sinha, Alan Black, and Justine Cassell. 2016a. Automatic recognition of conversational strategies in the service of a socially-aware dialog system. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 381–392, Los Angeles. Association for Computational Linguistics.

Ran Zhao, Tanmay Sinha, Alan W. Black, and Justine Cassell. 2016b. Socially-aware virtual agents: Automatically assessing dyadic rapport from temporal patterns of behavior. In *International conference on intelligent virtual agents*, page 218–233. Springer.

Caleb Ziems, Minzhi Li, Anthony Zhang, and Diyi Yang. 2022. Inducing positive perspectives with text reframing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3682–3700, Dublin, Ireland. Association for Computational Linguistics.

A Clauses to Turns

In our task formulation, a dialogue is composed of tutor-tutee turns. However, in the corpus considered for this study, the available annotations are at the clause[2] level. The choice of annotation unit was made because the annotation in hedges was part of a larger annotation campaign dedicated to the annotation of various conversational strategies (e.g., praise) at the clause level. This corpus contains 23 156 clauses, of which 21 192 contain non-hedges and 1 964 hedges. In order to obtain annotations at a turn level, we apply the simplest way to merge the hedge labels. If one or multiple clauses of one turn are annotated as hedges, this turn is labeled as a hedge.

B Metrics

BLEU (Papineni et al., 2002) calculates the word overlaps between reference and candidate utterances in n-grams (n=1, 2, 3). We do not assume that higher BLEU scores are equivalent to better task completion. Instead, BLEU is used to indicate that the generated utterances retain certain desired keywords.

ROUGE-L (Lin, 2004) supplements BLEU by computing the longest common subsequence of generated utterances and references, allowing it to compute overlap measures in longer utterances. To avoid generated utterances that are too long for the BLEU score, we use Rouge-L as a complementary metric.

CHRF (Popović, 2015) is comparable to BLEU; however, while BLEU is word-level, CHRF is character-level, based on character n-gram computation. Our transcribed dataset also shows some disfluencies and repetitions represented by individual characters. Therefore, we expect this metric to result in character-level overlap scores.

BERTScore (Zhang et al., 2020a) embeds the generated utterances and the reference with word vectors using the BERT model and computes pairwise cosine similarity for each generated word vector and each word in the reference, then the recall of the generated sequences is calculated. BERTScore is distinct from the previous two metrics in that it computes similarity across semantic space and has been shown to have a strong correlation with human judgment at the segment level.

BARTScore (Yuan et al., 2021) formulates the text generation evaluation as a text generation task from pretrained language models in an unsupervised fashion. When the generated text is better, the training model will get a higher score by converting the generated text to reference or source text. BART score can be applied to different evaluations (e.g., informativeness, coherence, and factuality).

Perplexity (Chen et al., 1998) calculates language model perplexity. Perplexity quantifies the level of uncertainty when an LM generates a new token.

C Implementation Details

The implementation of all models was based on the Transformer library[3], in addition, the Pytorch-Lightning[4] library was used for training control. We apply AdamW (Loshchilov and Hutter, 2018) as our optimizer with a learning rate $10e^{-}5$. All the models are trained with 10 epochs but with an Early-stopping mechanism on validation loss, which means when the validation loss remains for 2 epochs, the training will stop to prevent overfitting. We use the base version of the BART model, the small version of BlenderBot, and also the small version of DialoGPT. For the reranking method, we use beam search as our decoding strategy. To prevent repetition, we allow the 2 grams to oc-

[2] A clause consists of a subject and a verb and expresses a complete thought (Berry and Brizee, 2010).

[3] github.com/huggingface/transformers

[4] github.com/Lightning-AI/lightning

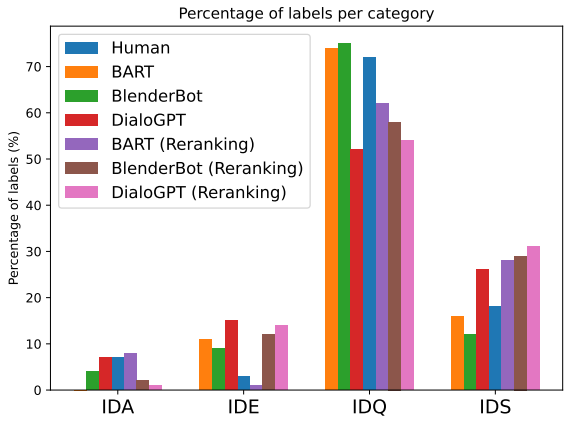

Figure 4: Hedge subcategories distribution in models' outputs compared with human. IDA: Apologizer; IDE: Extender; IDQ: Propositional hedges; IDS: Subjectivizer (as defined in Section 2.1)

cur only once, and the repetition penalty = 1.2 is also applied. All models were fine-tuned on an Nvidia Quadro RTX 8000 GPU. A complete configuration of the hyperparameters used for each model is reported in the GitHub repository with the code of the paper: `github.com/neuromaancer/hedge_generation`.

Moreover, we apply beam search for the decoding strategy, as it reduces the risk of missing hidden high-probability word sequences by retaining the n most likely words in each generation output and ultimately selecting the utterances with the highest overall probability. To avoid repeating the same subsequences, we apply a penalty to the repeated 2-gram unit. In terms of the size of the candidate pool, logically, the more candidates generated, the more chances that one of them is the right hedge strategy (i.e., hedge or non-hedge), so we fix our candidate pool size to 50, as a compromise between the likelihood of obtaining a hedge and the speed of generation.

D Figures

Figure 4: Hedge subcategories distribution in models' outputs compared with human.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 7: Limitation after the Section 6: Conclusion and future work

☑ A2. Did you discuss any potential risks of your work?
Section 8 Ethical Statement

☑ A3. Do the abstract and introduction summarize the paper's main claims?
In the abstract section

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☒ Did you use or create scientific artifacts?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
No response.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
No response.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
No response.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
No response.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
No response.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
No response.

C ☑ Did you run computational experiments?

Section 4: Experimental Setting

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 4: Experimental Setting

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 4: Experimental Setting

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 5: Results

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 4: Experimental Setting

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Section 3.2 Corpus and Section 4.4 Human Evaluation

☒ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
The instructions are described in other paper, the author used the dataset under a NDA. For the anonymity, we didn't cite these papers in the blind review session, but we will cite them in the final version of the paper.

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
4.4 Human Evaluation

☑ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
4.4 Human Evaluation

☑ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Section 3.2 Corpus

☑ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Section 3.2 Corpus