"You might think about slightly revising the title": identifying hedges in peer-tutoring interactions

Anonymous ACL submission

Abstract

Hedges play an important role in the management of conversational interaction. In peer-tutoring, they are notably used by tutors in dyads (pairs of interlocutors) experiencing low rapport to tone down the impact of instructions and negative feedback. Pursuing the objective of building a tutoring agent that manages rapport with students in order to improve learning, we used a multimodal peer-tutoring dataset to construct a computational framework for identifying hedges. We compared approaches relying on pre-trained resources with others that integrate insights from the social science literature. Our best performance involved a hybrid approach that outperforms the existing baseline while being easier to interpret. We employ a model explainability tool to explore the features that characterize hedges in peer-tutoring conversations, and we identify some novel features, and the benefits of such a hybrid model approach.

1 Introduction

Rapport, most simply defined as the "… relative harmony and smoothness of relations between people …" (Spencer-Oatey, 2005), has been shown to play a role in the success of activities as varied as psychotherapy (Leach, 2005) and survey interviewing (Lune and Berg, 2017). In peer-tutoring, rapport, as measured by the annotation of thin slices of video, has been shown to be beneficial for learning outcomes (Zhao et al., 2014; Sinha and Cassell, 2015). The level of rapport rises and falls with conversational strategies deployed by tutors and tutees at appropriate times, and as a function of the content of prior turns. These strategies include self-disclosure, referring to shared experience, and, on the part of tutors, giving instructions in an indirect manner. Some work has attempted to automatically detect these strategies in the service of intelligent tutors (Zhao et al., 2016a), but only a few strategies have been attempted. Other work has concentrated on a "social reasoning module" (Romero et al., 2017) to decide which strategies should be generated in a given context, but indirectness was not among the strategies targeted. In this paper, we focus on the automatic classification of one specific strategy that is particularly important for the tutoring domain, and therefore important for intelligent tutors: hedging, a sub-part of indirectness that "softens" what we say. This work is part of a larger research program with the long-term goal of generating indirectness behaviors for a tutoring agent.

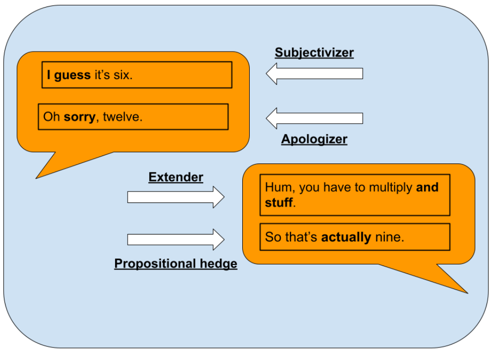

Figure 1: A mock conversation displaying each type of hedged formulation.

According to Brown and Levinson (1987), hedges are part of the linguistic tools that interlocutors use to produce politeness, by limiting the face threat to the interlocutor (basically by limiting the extent to which the interlocutor might experience embarrassment because of some kind of poor performance). An example is "that's *kind of* a wrong answer". Hedges are also found when speakers wish to avoid losing face themselves, for example when saying ("*I think* I *might* have to add 6."). Madaio et al. (2017) found that in a peer-tutoring task, when rapport between interlocutors is low, tutees attempted more problems and correctly solved more problems when their tutors hedged instructions, which likewise points towards a "mitigation of face threat" function. Hedges can also be associated with a nonverbal component, for example averted eye gaze during criticism (Burgoon and Koper, 1984). Hedges are not, however, always appropriate, as in "I *kind of think* it's raining today." when the interlocutors can both see rain (although it might be taken as humorous). These facts about hedges motivate a way to automatically detect them and, ultimately (although not in the current work) also generate them. In both cases we first have to be able to characterize them using interpretable linguistic features, which is what we address here. Thus, in the work described here, based on linguistic descriptions of hedges (Brown and Levinson, 1987; Fraser, 2010), we built a rule-based classifier. We show that this classifier in combination with additional multimodal interpretable context-dependent features significantly improves the performance of a machine learning model for hedges, compared to a less interpretable deep learning baseline from Goel et al. (2019) using word embeddings. We also relied on a machine learning model explanation tool (Lundberg and Lee, 2017) to investigate the linguistic features related to hedges in the context of peer-tutoring, primarily to see if we could discover surprising features that the classification model would associate to hedges in this context, and we describe those below. We release the code of the models described in the paper. [1]

2 Related work

Hedges: According to Fraser (2010), hedging is a rhetorical strategy that attenuates the strength of a statement. One way to produce an hedge is by altering the full semantic value of a particular expression through **Propositional hedges** (also called **Approximators** in Prince et al. (1982)), as in "You are *kind of* wrong," that reduce prototypicality (i.e accuracy of the correspondence between the proposition and the reality that the speaker seeks to describe). Propositional hedges are related to fuzzy language (Lakoff, 1975), and therefore to the production of vagueness (Williamson, 2002) and uncertainty (Vincze, 2014).

A second kind are **Relational Hedges** (also called **Shields** in Prince et al. (1982)), such as "*I think that* you are wrong." or "*The doctor wants you* to stop smoking.", conveying that the proposition is considered by the speaker as subjective. In a further sub-division, **Attribution Shields**, as in "The doctor *wants you* ...", the involvement of the speaker in the truth value of the proposition is not made explicit, which allows speakers not to take a stance. As described above, Madaio et al. (2017) found that tutors who showed lower rapport with their tutees used more hedged instructions (they also employed more positive feedback), however this was only the case for tutors with a greater belief in their ability to tutor. Tutees in this context did solve more problems correctly when their tutors hedged instructions. No effect of hedging was found for dyads (pairs of interlocutors) with greater social closeness. However, the authors did not look at the specific linguistic forms these teenagers used.

Rowland (2007) also describes the role that hedging plays in this age group, showing that students use both relational ("*I think that* John is smart.") and propositional ("John is *kind of* smart.") hedges for much the same shielding function of demonstrating uncertainty to save them from the risk of embarrassment if they are wrong. The author observed that teens used few **Adaptors** (*kind of*, *somewhat*) and preferred to use **Rounders** (*around*, *close to*). However, this study was performed with an adult and two children, possibly biasing the results due to the participation of the adult investigator. Hedges have been included in virtual tutoring agents before now. (Howard et al., 2015) integrated hedges in a tutor agent for undergraduates in CS, as a way to encourage the student to take initiatives. Hedges were also used as a way of integrating Brown and Levinson's politeness framework (Wang et al., 2008; Schneider et al., 2015) in virtual tutoring agents. Results were not broken out by strategy, but politeness in general was shown to positively influence motivation and learning, in certain conditions.

Computational methods for hedge detection: A number of studies have targeted the detection of hedges and uncertainty in text (Medlock and Briscoe, 2007; Ganter and Strube, 2009; Tang et al., 2010; Velldal, 2011; Szarvas et al., 2012), particularly following the CoNLL 2010 dataset release (Farkas et al., 2010). However, this work is not as related to hedges in conversation, as it focuses on a formal and academic language register (Hyland, 1998; Varttala, 1999). As noted by Prokofieva and Hirschberg (2014), the functions of hedges are domain- and genre-dependent, therefore this bias towards formality implies that the existing work

[1] https://github.com/AnonymousHedges/HedgeDetection

may not adapt well to the detection of hedges in conversation between teenagers. A consequence is that the existing work does not consider terms like "I think," since opinions rarely appear in an academic writing dataset. Instructions are also almost absent ("I think you have to add ten to both sides."), a strong limitation for the study of conversational hedges since it is in requests (including tutoring instructions) that indirect formulations mostly occur, according to Blum-Kulka (1987). Prokofieva and Hirschberg (2014) also note that it is difficult to detect hedges because the word patterns associated with them have other semantic and pragmatic functions: considering "I think that you have to add x to both sides." vs "I think that you are an idiot.", it is not clear that the second use of "I think that" is an hedge marker. They advocate using machine learning approaches to deal with the ambiguity of these markers. Working on a conversational dataset, Ulinski et al. (2018) built a computational system to assess speaker commitment (i.e. at which point the speaker seems convinced by the truth value of a statement), in particular by relying on a rule-based detection system for hedges. Compared to that work, our rule-based classification model is directly detecting hedge classes, and we employ the predictions of the rule-based model as a feature for stronger machine learning models, designed to lessen the impact of the imbalance between classes. We also consider **apologies** when they serve a mitigation function (we then call them **Apologizers**), as was done by the authors of our corpus, and we also use the term **subjectivizers** as defined below, to be able to compare directly with the previous work carried out on this corpus. As far as we know, only Goel et al. (2019) have worked with a peer-tutoring dataset (the one that we also use), and they achieved their best classification result by employing an Attention-CNN model, inspired by Adel and Schütze (2016).

3 Problem statement

We consider a set D of conversations $D = (c_1, c_2, ..., c_{|D|})$, where each conversation is composed of a sequence of independent syntactic clauses $c_i = (u_1, u_2, ..., u_M)$, where M is the number of clauses in the conversation. Note that two consecutive clauses can be produced by the same speaker. Each clause is associated with a unique label corresponding to the different hedge classes described in Table 1: $y_i \in C$ = {**Propositional Hedges**, **Apologizers**, **Subjectivizers**, **Not hedged**}. Finally, an utterance u_i can be represented as a vector of features $X = (x_1, x_2, ..., x_N)$, where N represents the number of features we used to describe a clause. Our first goal is to design a model that predicts correctly the label y_i associated to u_i. It can be understood as the following research question:

RQ1: "Which models and features can be used to automatically characterize hedges in a peer-tutoring interaction?"

Our second goal is to identify, for each hedge class, the set of features $F_{class} = \{f_k\}, k \in [1, N]$ sorted by feature importance in the classification of $class$. It corresponds to the following research question:

RQ2: "What are the most important linguistic features that characterize our hedge classes in a peer-tutoring setting?"

4 Methodology

4.1 Corpus

Data collection: The dialogue corpus used here was collected as part of a larger study on the effects of rapport-building on reciprocal peer tutoring. 24 American teenagers (mean age = 13.5, min = 12, max = 15), half male and half female, came to a lab where half of the participants were paired with a same-age, same-gender friend, and the other half with a stranger. The participants were assigned to a total of 12 dyads that alternated tutoring one another in linear algebra equation solving for 5 weekly hour-long sessions, for a total corpus of nearly 60 hours of face-to-face interactions. Each session was structured such that the students engaged in brief social chitchat in the beginning, then one of the students was randomly assigned to tutor the other for 20 minutes. They then engaged in another social period, and concluded with a second tutoring period where the other student was assigned the role of tutor. Audio and video data were recorded, transcribed, and segmented for clause-level dialogue annotation, providing nearly 24 000 clauses. Non-speech segments (notably fillers and laughter) were maintained. Because of temporal misalignment for parts of the corpus, many paraverbal phenomena, such as prosody, were unfortunately not available to us. Since our access to the dataset is covered by a Non-Disclosure Agreement, it cannot be released publicly. However the original experimenters' Institutional Review Board (IRB) approval allows us to view, annotate, and use the data to train models. This also allows us to provide a link to a pixelated video example in an online appendix to the paper.

Data annotation: This dataset was annotated by Madaio et al. (2017), using hedge classes derived from Rowland (2007) (see Table 1). Comparing the annotations with the classes mentioned in the related work section, **Subjectivizers** would correspond to **Relational hedges** (Fraser, 2010), **Propositional hedges** and **Extenders** correspond to **Approximators** (Prince et al., 1982) with the addition of some discourse markers such as *just*. **Apologizers** are mentioned as linguistic tools related to negative politeness in Brown and Levinson (1987). Krippendorff's alpha for all four codes was over 0.7 (denoting an acceptable inter-coder reliability according to Krippendorff (2004)). Only the task periods of the interactions were annotated. The dataset is widely imbalanced, with more than 90% of the utterances belonging to the **Not hedged** class. Utterances labeled with **Extenders** class were considered here as **Propositional hedges**, because the annotation of **Extenders** class was not precise and reliable enough and both classes carry a similar semantic function. Some other instances were reannotated when they clearly did not match the description of the class given in the coding manual provided by the original authors of the corpus.

4.2 Features

Label from rule-based classifier (Label RB): We use the class label predicted by the rule-based classifier described in Section 4.3 as a feature. Our hypothesis is that the machine learning model can use this information to counterbalance the class imbalance. To take into account the fact that some rules are more efficient than others, we weighted the class label resulting from the rule-based model by the precision of the rule that generated it.

Unigram and bigram: We count the number of occurrences of unigrams and bigrams of the corpus in each clause. We used the lemma of the words for unigrams and bigrams using the nltk lemmatizer (Loper, 2002) and selected unigrams and bigrams that occurred in the training dataset at least fifty times. The goal was to investigate, with a bottom-up approach, to what extent the use of certain words characterizes hedge classes in tutoring. In Section 5 we examine the overlap between these words and those *a priori* identified by the rules.

Part-of-speech (POS): Hedge classes seem to be associated with different syntactic patterns: for example, subjectivizers most often contain a personal pronoun followed by a verb, as in "I guess", "I believe", "I think". We therefore considered the number of occurrences of POS-Tag n-grams (n=1, 2, 3) as features. We used the spaCy POS-tagger and considered POS unigrams, bigrams and trigrams that occur at least 10 times in the training dataset.

LIWC: Linguistic Inquiry and Word Count (LIWC) (Pennebaker et al., 2015) is standard software for extracting the count of words belonging to specific psycho-social categories (*e.g.*, Emotions, Religion). It has been successfully used in the detection of conversational strategies (Zhao et al., 2016a). We therefore count the number of occurrences of all the 73 categories from LIWC.

Tutoring moves (TM): Intelligent tutoring systems rely on specific tutoring moves to successfully convey content (as do human tutors). We therefore looked at the link between the tutoring moves, as annotated in Madaio et al. (2017), and hedges. For tutors, these moves are (1) instructional directives and suggestions, (2) feedback, and (3) affirmations, mostly explicit reflections on their partners'comprehension, while for tutees, they are (1) questions, (2) feedbacks, and (3) affirmations, mostly tentative answers.

Nonverbal and paraverbal behaviors: As in Goel et al. (2019), we included the nonverbal and paraverbal behaviors that are related to hedges. Specifically, we consider laughter and smiles, that have been shown to be effective methods of mitigation (Warner-Garcia, 2014), cut-offs indicating self-repairs, fillers like "Um", gaze shifts (annotated as Gaze at Partner, Gaze at the math worksheet, and Gaze elsewhere), and head nods. Each feature was present twice in the feature vector, one time for each interlocutor. Inter-rater reliability for nonverbal behavior was 0.89 (as measured by Krippendorff's alpha) for eye gaze, 0.75 for smile count, 0.64 for smile duration and 0.99 for head nod. Laughter is also reported in the transcript at the word level. We separate behaviors from the tutor from that of the tutee. The collection process for these behaviors is detailed further in Zhao et al. (2016b).

The clause-level feature vector was normalized by the length of the clause (except for the rule-based label). This length was also added as a feature.

Class	Definition	Example
Subjectivizers	Words that reduce intensity or certainty	"So then I would divide by two."
Apologizers	Apologies used to soften direct speech acts	"Oh sorry six b."
Propositional hedges	Qualifying words to reduce intensity or certainty of utterances	"It's actually eight."
Extenders	Words used to indicate uncertainty by referring to vague categories	"It'll be the number x or whatever variable you have."

Table 1: Definition of the classes

Prop. hedges	Apologizers	Subjectivizers	Not hedged	Total
1073	138	592	21858	23661

Table 2: Distribution of the classes

Features name	Automatic extraction	Vector size
Rule-based label	Yes	4
Unigram	Yes	~250
Bigram	Yes	~250
POS	Yes	~1200
LIWC	Yes	73
Nonverbal	No	24
Tutoring moves	No	6
Total		~1800

Table 3: List of automatically extracted and manually annotated features with their size.

Table 3 presents an overview of the final feature vector.

4.3 Classification models

The classification models used are presented here according to their level of integration of external linguistic knowledge.

Rule-based model: On the basis of the annotation manual used to construct the dataset from Madaio et al. (2017), and with descriptions of hedges from Rowland (2007), Fraser (2010) and Brown and Levinson (1987), we constructed a rule-based classifier that matches regular expressions indicative of hedges. The rules are detailed in Table 7 in the Appendix.

LGBM: Since hedges are characterized by explicit lexical markers, we tested the assumption that a machine learning model with a knowledge-driven representation for clauses could compete with a BERT model in performance, while being much more interpretable. We relied on LightGBM, an ensemble of decision trees trained with gradient boosting (Ke et al., 2017). This model was selected because of its performance with small training datasets and because it can ignore uninformative features, but also for its training speed compared to alternative implementations of gradient boosting methods.

Multi-layer perceptron (MLP): As a simple baseline, we built a multi-layer perceptron using three sets of features: a pre-trained contextual representation of the clause (SentBERT; Reimers and Gurevych (2019)) ; the concatenation of this contextual representation of the clause and a rule-based label (not relying on the previous clauses) ; and finally the concatenation of all the features mentioned in section 4.2, without the contextualized representation.

LSTM over a sequence of clauses: Since we are working with conversational data, we also wanted to test whether taking into account the previous clauses helps to detect the type of hedge class in the next clause. Formally, we want to infer y_i using $y_i = \max_{y \in Classes} P(y|X(u_i), X(u_{i-1}), ..., X(u_{i-K}))$, where K is the number of previous clauses that the model will take into account. The MLP model presented above infers y_i using $y_i = \max_{y \in Classes} P(y|X(u_i))$, therefore a difference of performance between the two models would be a sign that using information from the previous clauses could help to detect the hedged formulation in the current clause. We tested a LSTM model with the same representations for clauses as for the MLP model.

CNN with attention: Goel et al. (2019) established their best performance on hedge detection using a CNN model with additive attention over word (and not clause) embeddings. Contrary to the MLP and LSTM models mentioned above, this model tries to infer y_i using $y_i = \max_{y \in Classes} P(y|g(w_0), g(w_1), ..., g(w_L))$, with L representing the maximum clause length we allow, and g representing a function that turns the word w_j, $j \in [0, L]$ into a vector representation (for more details, please see Adel and Schütze (2016)).

BERT: To benefit from deep semantic and contextual representations of the utterances, we also fine-tuned BERT (Devlin et al., 2018) on our classification task. BERT is a pre-trained Transformers encoder (Vaswani et al., 2017) that has significantly improved the state of the art on a number of NLP tasks, including sentiment analysis. It produces a contextual representation of each word in a sentence, making it capable of disambiguating the meaning of words like "think" or "just" that are representative of certain classes of hedges. BERT, however, is notably hard to interpret.

4.4 Analysis tools

Looking at which features improve the performance of our classification models tells us whether these features are informative or not, but does not explain how these features are used by the models to make a given prediction. We therefore produced a complementary analysis using an interpretability tool. As demonstrated by (Lundberg and Lee, 2017), LightGBM internal feature importance scores are inconsistent with both the model behavior and human intuition, so we instead used a model-agnostic tool. SHAP (Lundberg and Lee, 2017) assigns to each feature an importance value (called Shapley values) for a particular prediction depending on the extent of its contribution (a detailed introduction to Shapley values and SHAP can be found in Molnar (2020)). SHAP is a model-agnostic framework, therefore the values associated with a set of features can be compared across models. It should be noted that SHAP produces explanations on a case-by-case basis, therefore it can both provide local and global explanations. For the Gradient Boosting model, we use an adapted version of SHAP (Lundberg et al., 2018), called TreeSHAP.

5 Experiments and results

5.1 Experimental setting

To detect the best set of features, we used LightGBM and proceeded incrementally, by adding the group of features we thought to be most likely associated with hedges. We did not consider the risk of relying on a sub-optimal set of features through this procedure because of the strong ability of LightGBM to ignore uninformative features. We use this incremental approach as a way to test our intuition about the performativity of groups of features (*i.e.* does adding a feature improve the performance of the model) with regard to the task of classification. To compare our models, we trained them on the 4-class task, and looked at the average of the weighted F1-scores for the three hedge classes (*i.e.* how well the models infer minority classes) that we report here as "3-classes", and at the average of the weighted F1-scores for the 4 classes, that we report as "4-classes". Details of the hyperparameters and experimental settings are provided in Appendix A.

5.2 Model comparison and feature analysis

Overall results: Table 4 presents the results obtained by the 6 models presented in Section 4.3 for the multi-class problem. Best performance (F1-score of 74.4) is obtained with LightGBM leveraging all the features. In the appendix (see Table 8 and Table 9) we indicate the confidence intervals to represent the significance of the differences between the models.

First, and perhaps surprisingly, we notice that the use of "Knowledge-Driven" features based on rules built from linguistic knowledge of hedges in the LightGBM model outperforms the use of pre-trained embeddings within a fine-tuned BERT model (74.4 vs. 65.9), and in the neural baseline from (Goel et al., 2019) (74.4 vs 58.9).

The low scores obtained by the LGBM, LSTM and MLP models with pre-trained sentence embeddings versus Knowledge-Driven features might signal that the word patterns characterizing hedges are not salient in these representations (i.e. the distance between "*I think* you should add 5." and "You should add 5." is short.). KD Features seem to provide a better separability of the classes. The combination of KD features and Pre-trained embeddings does not significantly improve the performance of the models compared to the KD Features only, which suggests that the information from the Pre-trained embeddings is redundant with the one from the KD Features. This result may be due to the high dimensionality of the input vector (868 with PCA on the KD Features; 2500 otherwise).

A second finding is that the use of gradient boosting models on top of rule-based classifiers better models the hedge classes. The other machine learning models did not prove to be as effective, except for BERT.

Feature analysis using LightGBM: Using the best performing model, Table 5 shows the role of each feature set in the prediction task. The significance of the differences is shown in Table 10 and Table 11. Compared to the rule-based model, the introduction of n-grams significantly improved the performance of our classifier, suggesting that some lexical and syntactic information describing the hedge classes was not present in the rule-based model. Looking at Table 5, we do not observe significant differ-

Models	KD Feat. (KDF)	Pre-Trained Emb. (PTE)	KDF + PTE
Rule-based (3-classes)	61.9	∅	∅
MLP (3-classes)	62.6 (1.7)	36.0 (2.8)	62.1 (0.8)
Attention-CNN (3-classes)	∅	58.9 (2.4)	∅
LSTM (3-classes)	59.8 (8.9)	34.3 (5.8)	61.0 (3.3)
BERT (3-classes)	∅	**65.9 (2.9)**	∅
LGBM (3-classes)	**74.4 (1.3)**	43.0 (0.6)	**73.8 (1.9)**
Rule-based (4-classes)	94.3	∅	∅
MLP (4-classes)	94.3 (0.3)	90.9 (0.3)	94.1 (0.2)
Attention-CNN (4-classes)	∅	94.3 (0.2)	∅
LSTM (4-classes)	93.0 (2.7)	89.8 (1.8)	93.7 (0.7)
BERT (4-classes)	∅	**94.5 (0.8)**	∅
LGBM (4-classes)	**96.3 (0.2)**	92.1 (0.1)	**96.2 (0.3)**

Table 4: Averaged weighted F1-scores (and standard deviation) for the three minority classes and for the 4 classes, for all models. "KD" stands for "Knowledge-Driven", meaning that the features are derived from lexicon, n-gram models and annotations.

ences between the LGBM model using only the label rule based + (1-grams and 2-grams) and the models incorporating more features. To our surprise, neither the tutoring moves nor the nonverbal features significantly improved the performance of the model. The 2 features were included to index the specific peer tutoring context of these hedges, so this indicates that in future work we might wish to apply the current model to another context of use to see if this model of hedges is more generally applicable than we originally thought. By combining this result with the better performance of the model using Knowledge-Driven (*i.e.* explicit) features compared to pre-trained embeddings, it would seem that hedged formulations is above all a lexical phenomena (*i.e.* produced by specific lexical elements).

5.3 In-depth analysis of the informative features

We trained the SHAP explanation models on LightGBM with all features. The most informative features (in absolute value) for each class are shown in Table 6, and the plots by class are presented in the Appendix. The most important features seem to be the rule-based labels, which appear in at least the third position for three classes (see Table 6), and in the first position for **Propositional Hedges** and **Not hedged** classes. Unigrams (*Oh*, *Sorry*, *just*, *Would*, and *I*) are also present in the 5 top-ranked features. This confirms the findings mentioned in related work for the characterization of the different hedge classes (*just* with **Propositional Hedges**, *sorry* with **Apologizer**, *I* with **Subjectivizers**). The presence of *Oh* also has high importance for the characterization of **Apologizer** (n=7, see Figure 4), as illustrated in examples such as "*Oh* sorry, that's nine.". We note that the occurrences of "*Oh sorry*" as a clause were excluded by our rule-based model because they do not correspond to an apologizer (they cannot mitigate the content of a proposition if there is no proposition associated). This example illustrates the interest of a machine learning model approach to disambiguate the function of conventional non-propositional phrases like "*Oh sorry*".

In addition, SHAP highlights the importance of novel features that were not already identified by the literature: *(i)* what LIWC classifies as **informal words** but that are mostly interjections like *ah* and *oh* are strongly associated with **Apologizer**, as are disfluencies; *(ii)* the use of **POS tags** seems to be very relevant for characterizing the different classes (POS tag features[2] occur in the top-ranked features of all the classes (see Figures in the Appendix). It means that there are some recurring syntactic patterns in each class; *(iii)* Regarding the **utterance size**, a clause shorter than the mean is associated with **directness** while a longer clause suggests that it contains a **Subjectivizer (n=4)**; *(iv)* Tutoring moves are not strong predictors of any classes: "Affirmation from tutor" is the only feature appearing as a predictor of Propositional hedges (n=19). This is consistent with the feature analysis in Table 5, suggesting that tutoring moves do not significantly improve the performance of the classifier; *(v)* **Nonverbal behaviors** do not appear as important features for the classification. This is coherent with results from (Goel et al., 2019). Note that prosody might play a role in detecting instructions that trail off, but, as described, paraverbal features were not available; *(vi) Would* plays an important role in the data classification process, as it is strongly associated to **Propositional hedges** (n=2). We can see in Figure 6 that if the clauses containing the form (PRONOUN, VERB) are more likely to be **Subjectivizers** (n=13), some clauses are positively associated with the class because they **do not** contain "*I would*". It may mean that "*I would*" is a 2-gram that corresponds to the (PRONOUN, VERB) pattern, but that is not as much associated to the class as the rest of the instances of that pattern. It is interesting to note that, when designing the rule-based classifier, we saw it reach a limit in performance when we started to include

[2] Note that there is strong redundancy between some features of LIWC and the spaCy POS tagger that both produce a "Pronoun" category, using a lexicon in the first case, and a neural inference in the second.

Models	Label RB	+ 1-gram and 2-gram	+ POS	+ LIWC	+ TM	+ Nonverbal
3-classes	63.2 (1.3)	75.0 (0.7)	74.0 (1.0)	74.0 (1.4)	75.1 (1.6)	74.4 (1.3)
4-classes	94.6 (0.2)	96.3 (0.1)	96.2 (0.2)	96.2 (0.1)	96.3 (0.3)	96.3 (0.2)

Table 5: Averaged weighted F1-scores for the three classes of hedges and the four classes, with an additive integration of features in the LightGBM model. The standard deviation is computed across five folds.

Rank	Apologizer	Subjectivizers	Prop. Hedges	Not hedged
1	Affect (LIWC)	Yeah	Class label	Class label
2	Disfluencies (LIWC)	I	Would	Would
3	Sorry	Class label	Just	Yeah
4	Sad (LIWC)	Clause length (higher)	Function word (LIWC)	Insight (LIWC)
5	Focus present (LIWC)	Cognitive process (LIWC)	Actually	(Interject°, Interject°)

Table 6: Most important clause-level features for LightGBM according to the SHAP analysis.

Would in our regular expression patterns, probably because the form is hard to disambiguate for a deterministic system.

While exploring the Shapley values associated to each clause, we observed that features like tutoring moves are extremely informative for a very small number of clauses (therefore not significantly influencing the overall performance of the prediction), and more or less not informative for the rest. Inferring the global importance of a feature as a mean across the shapley values in the dataset may not be the only way to explore the behavior of gradient boosting methods. It might be more useful to cluster clauses based on the importance that SHAP gives to that feature in its classification, as this could help discover sub-classes of hedges that are differentiated from the rest by their interaction with a specific feature (in the way that some **Apologizers** are characterized by an "oh"). We also note that the explanation model is sensitive to spurious correlations in the dataset, caused by the small representation of some class: for example, "nine" is a positive predictor (n=8) of **Apologizers**.

6 Conclusion and future work

Through our classification performance experiments, we showed that it is possible to use machine learning methods to diminish the ambiguity of hedges, and that the hybrid approach of using rule-based label features derived from social science (including linguistics) literature within a machine learning model helped significantly to increase the model's performance. Nonverbal behaviors and tutoring moves did not provide information at the sentence level; both the performance of the model and the feature contribution analysis suggested that their impact on the model output was not strong. This is consistent with results from Goel et al. (2019). However, in future work we would like to investigate the potential of multimodal patterns when we are able to better model sequentiality (*e.g.*, negative feedback followed by a smile). Regarding the SHAP analysis, most of the features that are considered as important are coherent with the definition of the classes (*I* for subjectivizers, *sorry* for apologizers, *just* for propositional hedges). However, we discovered that features like utterance size can serve as indicator of certain classes of hedges. A limitation of SHAP is that it makes a feature independence assumption, which prompts the explanatory model to underestimate the importance of redundant features (like pronouns in our work). In the future we will explore explanatory models capable of taking into account the correlation between features in the dataset like SAGE (Covert et al., 2020), but suited for very imbalanced datasets. In the domain of peer-tutoring, we would like to be able to further test the link between hedges and rapport, and the link between hedges and learning gains in the subject being tutored. As noted above, this kind of study requires a fine-grained control of the language produced by one of the interlocutors, which is difficult to control in a human-human experience. The hedge classifier can be used not just to classify, but also to work towards improving their generation for tutor agents. In future work we will explore using the classifier to re-rank generation outputs, taking advantage of the recurring syntactic patterns (see *(ii)* in Section 5.3) to improving the generation process of hedges, and re-generating clauses that don't contain one of these syntactic patterns.

References

Heike Adel and Hinrich Schütze. 2016. Exploring different dimensions of attention for uncertainty detection. *arXiv preprint arXiv:1612.06549*.

Shoshana Blum-Kulka. 1987. Indirectness and politeness in requests: Same or different? *Journal of pragmatics*, 11(2):131–146.

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

Judee K Burgoon and Randall J Koper. 1984. Nonverbal and relational communication associated with reticence. *Human Communication Research*, 10(4):601–626.

Ian Covert, Scott Lundberg, and Su-In Lee. 2020. Understanding global feature contributions with additive importance measures. *arXiv preprint arXiv:2004.00668*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Richárd Farkas, Veronika Vincze, György Móra, János Csirik, and György Szarvas. 2010. The conll-2010 shared task: learning to detect hedges and their scope in natural language text. In *Proceedings of the fourteenth conference on computational natural language learning–Shared task*, pages 1–12.

Bruce Fraser. 2010. Pragmatic competence: The case of hedging. *New approaches to hedging*, 1534.

Viola Ganter and Michael Strube. 2009. Finding hedges by chasing weasels: Hedge detection using wikipedia tags and shallow linguistic features. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 173–176.

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

Cynthia Howard, Pamela W. Jordan, Barbara Maria Di Eugenio, and Sandra Katz. 2015. Shifting the load: a peer dialogue agent that encourages its human collaborator to contribute more to problem solving. *International Journal of Artificial Intelligence in Education*, 27:101–129.

Ken Hyland. 1998. *Hedging in scientific research articles*, volume 54. John Benjamins Publishing.

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*, 30:3146–3154.

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 Leach. 2005. Rapport: A key to treatment success. *Complementary therapies in clinical practice*, 11:262–5.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Scott M Lundberg, Gabriel G Erion, and Su-In Lee. 2018. Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems*, pages 4768–4777.

Howard Lune and Bruce L Berg. 2017. *Qualitative research methods for the social sciences*. Pearson.

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.

Ben Medlock and Ted Briscoe. 2007. Weakly supervised learning for hedge classification in scientific literature. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 992–999.

Christoph Molnar. 2020. *Interpretable machine learning*. Lulu. com.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037.

James W Pennebaker, Ryan L Boyd, Kayla Jordan, and Kate Blackburn. 2015. The development and psychometric properties of liwc2015. Technical report.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

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

Anna Prokofieva and Julia Hirschberg. 2014. Hedging and speaker commitment. In *5th Intl. Workshop on Emotion, Social Signals, Sentiment & Linked Open Data, Reykjavik, Iceland*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Oscar J Romero, Ran Zhao, and Justine Cassell. 2017. Cognitive-inspired conversational-strategy reasoner for socially-aware agents. In *IJCAI*, pages 3807–3813.

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

Sascha Schneider, Steve Nebel, Simon Pradel, and Günter Daniel Rey. 2015. Mind your ps and qs! how polite instructions affect learning with multimedia. *Computers in Human Behavior*, 51:546–555.

Tanmay Sinha and Justine Cassell. 2015. We click, we align, we learn: Impact of influence and convergence processes on student learning and rapport building. In *Proceedings of the 1st Workshop on Modeling INTERPERsonal SynchrONy And InfLuence*, INTERPERSONAL '15, page 13–20, New York, NY, USA. Association for Computing Machinery.

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

György Szarvas, Veronika Vincze, Richárd Farkas, György Móra, and Iryna Gurevych. 2012. Cross-genre and cross-domain detection of semantic uncertainty. *Computational Linguistics*, 38(2):335–367.

Buzhou Tang, Xiaolong Wang, Xuan Wang, Bo Yuan, and Shixi Fan. 2010. A cascade method for detecting hedges and their scope in natural language text. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning–Shared Task*, pages 13–17.

Morgan Ulinski, Seth Benjamin, and Julia Hirschberg. 2018. Using hedge detection to improve committed belief tagging. In *Proceedings of the Workshop on Computational Semantics beyond Events and Roles*, pages 1–5.

Teppo Varttala. 1999. Remarks on the communicative functions of hedging in popular scientific and specialist research articles on medicine. *English for specific purposes*, 18(2):177–200.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Erik Velldal. 2011. Predicting speculation: a simple disambiguation approach to hedge detection in biomedical literature. *Journal of Biomedical Semantics*, 2(5):1–14.

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

Ning Wang, W Lewis Johnson, Richard E Mayer, Paola Rizzo, Erin Shaw, and Heather Collins. 2008. The politeness effect: Pedagogical agents and learning outcomes. *International journal of human-computer studies*, 66(2):98–112.

Shawn Warner-Garcia. 2014. Laughing when nothing's funny: The pragmatic use of coping laughter in the negotiation of conversational disagreement. *Pragmatics*, 24(1):157–180.

Timothy Williamson. 2002. *Vagueness*. Routledge.

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 W 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.

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*, pages 218–233. Springer.

A Additional information on the experimental settings

We used PyTorch (Paszke et al., 2019) to implement the neural models. For each set of features, hyperparameters were selected using Optuna (Akiba, 2019), a parameter search framework. We re-implemented the Attention-CNN with Glove (Pennington et al., 2014) 300-D words embeddings as the vector representation. For each models, the results are cross-validated using 5 folds (we chose 5 instead of 10 to avoid having folds with too few samples per class). We corrected the loss function for class imbalance to force the model to adapt more to the less frequent classes. The strength of this correction depended on the model, and was selected because it provided a satisfying compromise between favoring recall and precision in the classification results of that model. For LightGBM, a "square root of the square root of the inverse class proportion" correction was selected. Neural models were trained using AdamW as an optimizer (Loshchilov and Hutter, 2017), and used a reduced feature vector, obtained with the application of PCA (d_{init} = 1800; d = 100 ; 99.8 % of the information is conserved). No significant performance differences were observed between the original vector and the reduced vector for training the models. To compute the SHAP values mentioned in the paper, we kept one split to perform the 5-split of the dataset, and leave 1 split to validate and early stop the model, in order to avoid overfitting. A complete configuration of hyperparameters used for each model is reported in the GitHub repository with the code of the paper: https://github.com/AnonymousHedges/HedgeDetection. The BERT model was fine-tuned on a Nvidia Quadro RTX 8000 GPU.

B Tables

<table>
<tr><th>Class</th><th>Rule (regexp)</th></tr>
<tr><td>Subj.</td><td>(?!what).*(i|we) ?(don’t|didn’t|did)? ?(not)?
(guess|guessed|thought|think|believe|believed|suppose|supposed)
?(whether|if|is|that|it|this)?.*</td></tr>
<tr><td>Subj.</td><td>.*(i|i’m|we) ?(was|am|wasn’t)? ?(not)? (sure|certain).*</td></tr>
<tr><td>Subj.</td><td>.*(i feel like you).*</td></tr>
<tr><td>Subj.</td><td>.*(you (might|may) (believe|think)).*</td></tr>
<tr><td>Subj.</td><td>.*(according to|presumably).*</td></tr>
<tr><td>Subj.</td><td>.*(i|you|we) have to (check|look|verify).*</td></tr>
<tr><td>Subj.</td><td>.*(if i’m not wrong|if i’m right|if that’s true).*</td></tr>
<tr><td>Subj.</td><td>.*(unless i).*</td></tr>
<tr><td>Apol.</td><td>.*(i’m|i|we’re) (am|are)? ?(apologize|sorry).*</td></tr>
<tr><td>Apol.</td><td>(?!.*(be|been|was) like excuse me)((excuse me|sorry)[w ,’]+|[w ,’]+(excuse me|sorry))</td></tr>
<tr><td>Prop.</td><td>.*(just|a little|maybe|actually|sort of|kind of|pretty
much|somewhat|exactly|almost|little bit|quite|
regular|regularly|actually|almost|as it were|basically|
probably|can be view as|crypto-|especially|essentially|
exceptionally|for the most part|in a manner of speaking|
in a real sense|in a sense|in a way|largely|literally|
loosely speaking|kinda|more or less|mostly|often|
on the tall side|par excellence|particularly|
pretty much|principally|pseudo-|quintessentially|
relatively|roughly|so to say|strictly speaking|
technically|typically|virtually|approximately|
something between|essentially|only).*</td></tr>
<tr><td>Prop.</td><td>.*(i|i’m|you|it’s) (am|are) (apparently|surely)[,]?.*</td></tr>
<tr><td>Prop.</td><td>.*(it) (looks|seems|appears)[,]?.*", ".* (or|and) (that|something|stuff|so forth)</td></tr>
</table>

Table 7: Regexp rules used for the classifier.

Models	RB	MLP (KDF)	MLP (PTE)	MLP (KDF/PTE)	CNN (PTE)	LSTM (KDF)	LSTM(PTE)	LSTM (KDF/PTE)	BERT (PTE)	LGBM (KDF)	LGBM (PTE)	LGBM (KDF/PTE)
Rule-based		No	Yes	No	Yes	No	Yes	No	Yes	Yes	Yes	Yes
MLP (KDF)	No		Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
MLP (PTE)	Yes	Yes		Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes
MLP (KDF + PTE)	No	No	Yes		Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
Attention-CNN (PTE)	Yes	Yes	Yes	Yes		No	Yes	No	Yes	Yes	Yes	Yes
LSTM (KDF)	No	Yes	Yes	Yes	No		Yes	No	Yes	Yes	Yes	Yes
LSTM(PTE)	Yes	Yes	No	Yes	Yes	Yes		Yes	Yes	Yes	Yes	Yes
LSTM (KDF + PTE)	No	No	Yes	No	No	No	Yes		Yes	Yes	Yes	Yes
BERT (PTE)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes		Yes	Yes	Yes
LGBM (KDF)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes		Yes	No
LGBM (PTE)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes		Yes
LGBM (KDF + PTE)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	

Table 8: Significance table for the 3-classes part of Table 4. "Yes" means that the difference is statistically significant.

Models	RB	MLP (KDF)	MLP (PTE)	MLP (KDF/PTE)	CNN (PTE)	LSTM (KDF)	LSTM(PTE)	LSTM (KDF/PTE)	BERT (PTE)	LGBM (KDF)	LGBM (PTE)	LGBM (KDF/PTE)
Rule-based		No	Yes	No	No	Yes	Yes	Yes	No	Yes	Yes	Yes
MLP (KDF)	No		Yes	No	No	Yes	Yes	Yes	No	Yes	Yes	Yes
MLP (PTE)	Yes	Yes		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
MLP (KDF + PTE)	No	No	Yes		No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Attention-CNN (PTE)	No	No	Yes	No		Yes	Yes	Yes	No	Yes	Yes	Yes
LSTM (KDF)	Yes	Yes	Yes	Yes	Yes		Yes	Yes	Yes	Yes	Yes	Yes
LSTM(PTE)	Yes	Yes	Yes	Yes	Yes	Yes		Yes	Yes	Yes	Yes	Yes
LSTM (KDF + PTE)	Yes	Yes	Yes	Yes	Yes	Yes	Yes		Yes	Yes	Yes	Yes
BERT (PTE)	No	No	Yes	Yes	No	Yes	Yes	Yes		Yes	Yes	Yes
LGBM (KDF)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes		Yes	No
LGBM (PTE)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes		Yes
LGBM (KDF + PTE)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	

Table 9: Significance table for the 4-classes part of Table 4. "Yes" means that the difference is statistically significant.

Models	Label RB	+ 1-gram and 2-gram	+ POS	+ LIWC	+ TM	+ Nonverbal
Label RB		Yes	Yes	Yes	Yes	Yes
+ 1-gram and 2-gram	Yes		No	No	No	No
+ POS	Yes	No		No	No	No
+ LIWC	Yes	No	No		No	No
+ TM	Yes	No	No	No		No
+ Nonverbal	Yes	No	No	No	No	

Table 10: Significance table for the 3-classes part of Table 5. "Yes" means that the difference is statistically significant.

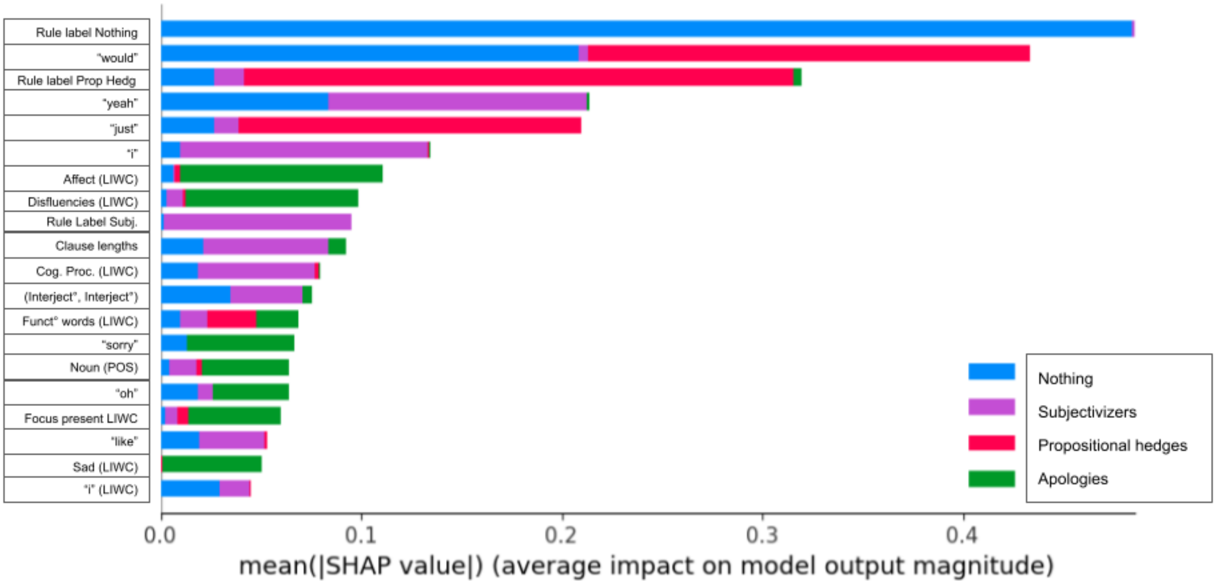

Figure 2: Absolute averaged feature contribution, as indicated by SHAP. The longer the bar is for one color, the more the feature is associated with the class represented by that color.

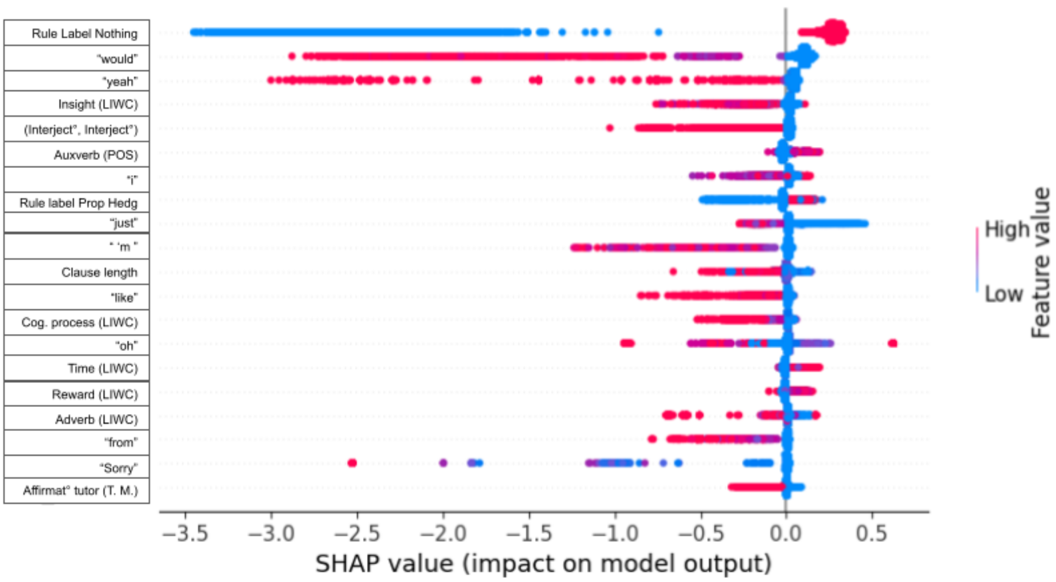

Figure 3: Averaged contribution of features to the detection of the "Not indirect" class, as indicated by SHAP. Each dot corresponds to a classified clause. A red dot indicates that the feature is present in the clause, while a blue dot indicates that the feature is absent. The farther on the right the dot is, the more the feature contributed to its classification as a hedge.

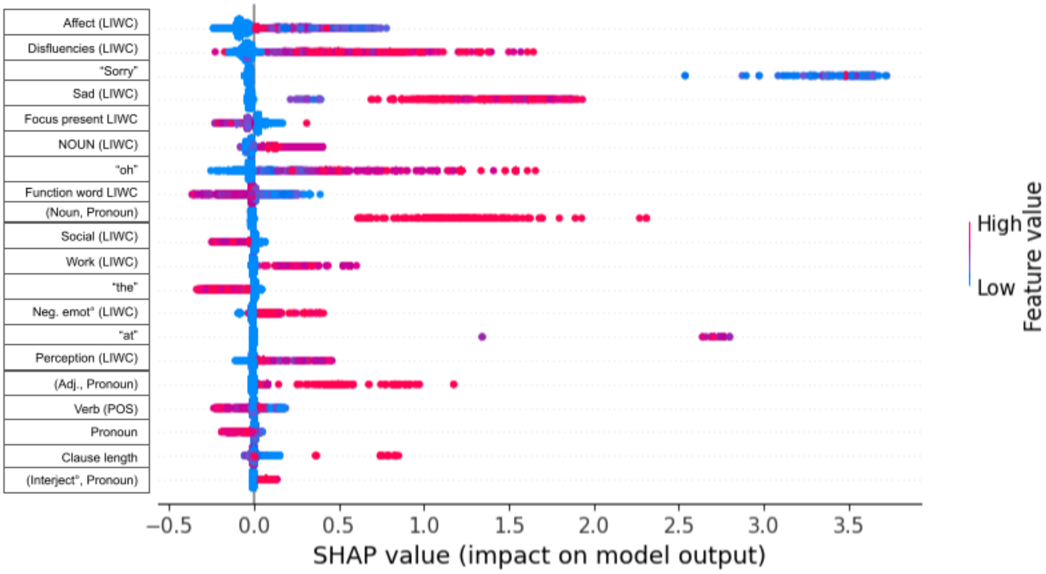

Figure 4: Averaged contribution of features to the detection of "Apologizers", as indicated by SHAP.

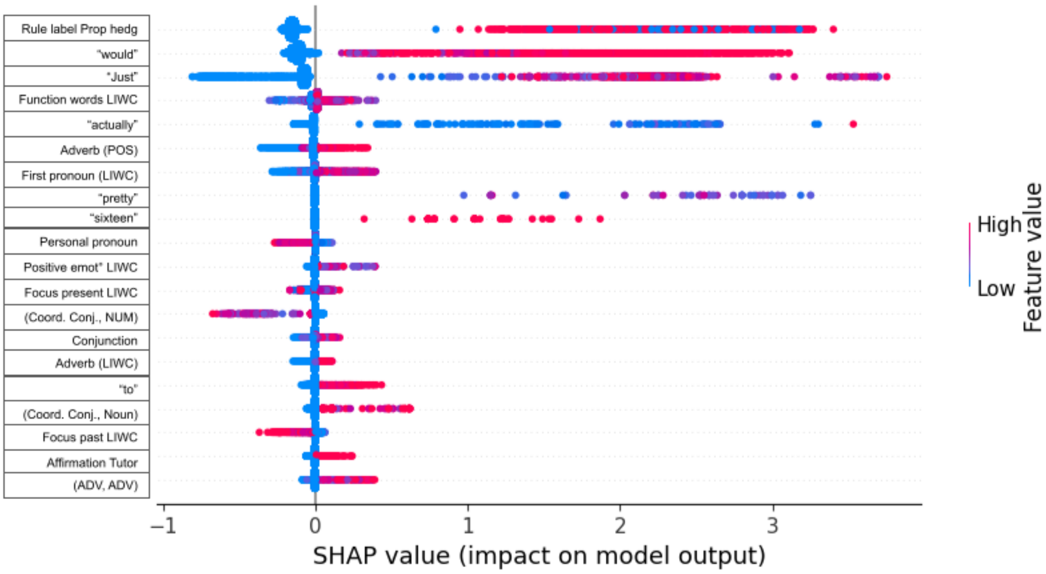

Figure 5: Averaged contribution of features to the detection of "Propositional hedges", as indicated by SHAP.

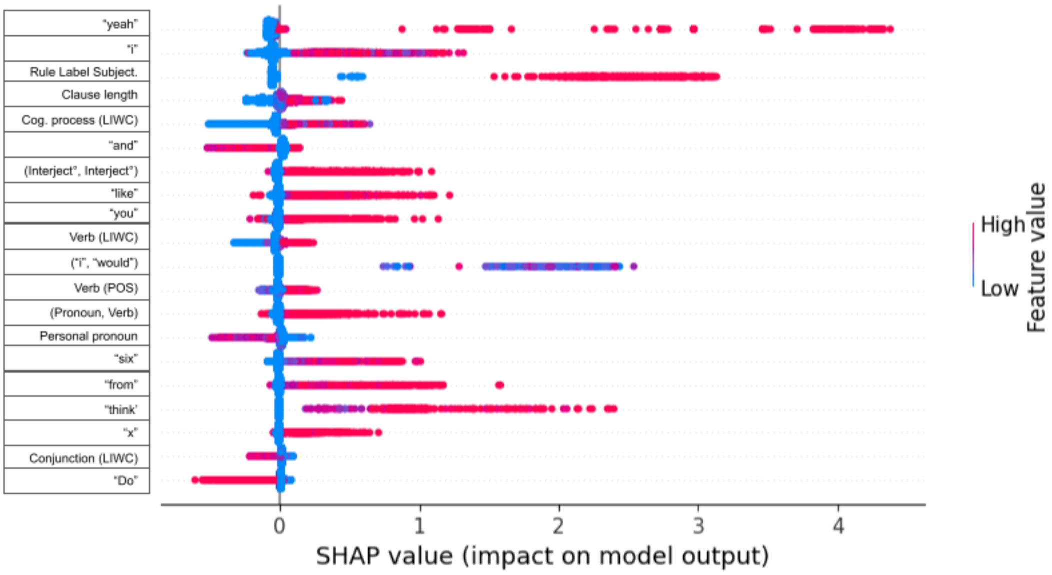

Figure 6: Averaged contribution of features to the detection of "Subjectivizers", as indicated by SHAP.

Models	Label RB	+ 1-gram and 2-gram	+ POS	+ LIWC	+ TM	+ Nonverbal
Label RB		Yes	Yes	Yes	Yes	Yes
+ 1-gram and 2-gram	Yes		No	No	No	No
+ POS	Yes	No		No	No	No
+ LIWC	Yes	No	No		No	No
+ TM	Yes	No	No	No		No
+ Nonverbal	Yes	No	No	No	No	

Table 11: Significance table for the 4-classes part of Table 5. "Yes" means that the difference is statistically significant.