"Are you kidding me?": Detecting Unpalatable Questions on Reddit

Sunyam Bagga
.txtLAB
McGill University
sunyam.bagga@mcgill.ca

Andrew Piper
Dept. of Languages, Literatures, & Cultures
McGill University
andrew.piper@mcgill.ca

Derek Ruths
School of Computer Science
McGill University
derek.ruths@mcgill.ca

Abstract

Abusive language in online discourse negatively affects a large number of social media users. Many computational methods have been proposed to address this issue of online abuse. The existing work, however, tends to focus on detecting the more explicit forms of abuse leaving the subtler forms of abuse largely untouched. Our work addresses this gap by making three core contributions. First, inspired by the theory of impoliteness, we propose a novel task of detecting a subtler form of abuse, namely *unpalatable questions*. Second, we publish a context-aware dataset for the task using data from a diverse set of Reddit communities. Third, we implement a wide array of learning models and also investigate the benefits of incorporating conversational context into computational models. Our results show that modeling subtle abuse is feasible but difficult due to the language involved being highly nuanced and context-sensitive. We hope that future research in the field will address such subtle forms of abuse since their harm currently passes unnoticed through existing detection systems.

1 Introduction

Abusive language and other antisocial behaviour is omnipresent in online discourse. According to a recent survey, 41% of Americans have personally experienced some form of online harassment (Duggan, 2017). To counter abusive behaviour online, different social media platforms implement their own mechanisms such as content moderation, muting or blocking users from posting etc. It is, however, infeasible to *manually* moderate online communities due to the sheer enormity of content produced every day – Twitter, for example, receives over 500 million tweets per day. Manual moderation in such a scenario would require humans to read millions of tweets daily which would take an impractical amount of time and other resources. Consequently, many computational models have been proposed by the Natural Language Processing (NLP) community to detect online abuse and facilitate automatic content moderation.

Abuse is an umbrella term which can cover several types of negative expressions. There exists a plethora of abuse detection studies employing different terminology: personal attacks (Wulczyn et al., 2017), bullying (Dadvar et al., 2013; Chatzakou et al., 2017), hate speech (Warner and Hirschberg, 2012; Davidson et al., 2017; Djuric et al., 2015; Gao and Huang, 2017), nastiness (Samghabadi et al., 2017), harassment (Golbeck et al., 2017; Yin et al., 2009), hostility (Liu et al., 2018), racism or sexism (Waseem and Hovy, 2016), abusive language (Nobata et al., 2016), aggression (Caines et al., 2018), and others. However, extant work in abuse detection has largely focused on detecting overt abuse ignoring the more subtle forms of abuse which can be just as damaging. This is also noted in a recent survey calling on the NLP community to rethink and expand what constitutes abuse (Jurgens et al., 2019).

In this work, we make three contributions to address this gap in the literature. First, inspired from the theory of linguistic impoliteness, we propose a novel task of detecting a subtler form of abuse called *unpalatable questions* (UQ). It is one of the conventionalized impoliteness formulae introduced by Culpeper (2010). We define the UQ task as detecting a negatively phrased question designed to antagonise its recipient in online discourse.

Second, we collect, annotate, and make publicly available a context-aware dataset for the UQ task.[1] The data comes from a diverse set of online communities (or *subreddits*) on the popular social media site Reddit. Most existing datasets used in abuse

[1] We make our dataset and code publicly available at https://github.com/networkdynamics/unpalatable-questions.

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pages 2083–2099
April 19 - 23, 2021.

detection (Wulczyn et al., 2017; Waseem and Hovy, 2016; Davidson et al., 2017; Founta et al., 2018; Golbeck et al., 2017) only include annotations for stand-alone comments or tweets. In comparison, we explicitly consider conversational context during annotation and preserve contextual information in our dataset (see Section 4.4 for a detailed comparison).

A major limitation of existing abuse detection studies – also pointed out in (Castelle, 2018; Mishra et al., 2019; Gao and Huang, 2017) – is that a comment is treated as a single-utterance in isolation, ignoring any conversational context provided by other comments in the discussion. This is problematic since abuse is inherently contextual, and it becomes a major issue when working with subtler forms of abuse such as unpalatable questions. To this end, our third contribution is that we implement a wide array of learning models to detect unpalatable questions and investigate the benefits of incorporating conversational context in our computational models.

2 What is an Unpalatable Question?

We adopt the term *unpalatable question* (UQ) from the conventionalised impoliteness formulae introduced by Culpeper (2010). Although Culpeper did not formally define UQ, several examples were laid out: 'why do you make my life impossible?', 'which lie are you telling me?', 'what's gone wrong now?'. We find that UQs tend to be rhetorical in nature in that they are usually asked not to elicit an answer but to make a point. In particular, they have a close resemblance to *epiplexis*: a type of rhetorical question which is asked not to elicit information but to reproach, upbraid, or rebuke (Zimmerman, 2005). This can be seen in the examples listed in Table 1, where the questions are asked to shame the interlocutor for adopting a particular point of view and are often insults asked as questions. For our task, we define an *unpalatable question* as a negatively phrased question designed to antagonise its recipient.

Why UQ? Jurgens et al. (2019) outline a spectrum of abusive behaviour highlighting that existing work only focuses on overt abuse ignoring both the subtler forms and extreme behaviours. As can be seen in Figure 1, our task of detecting a subtler form of abuse is a step towards addressing this gap. Moreover, studies in linguistics show that being asked an unpalatable question puts the recipient in a vulnerable position to receive further verbal attacks (Bousfield, 2007; Wijayanto et al., 2017).

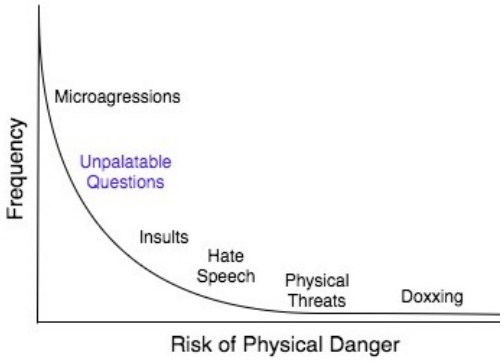


Figure 1: This figure, taken from (Jurgens et al., 2019), illustrates where *unpalatable questions* fit in a hypothetical spectrum of online abuse.

3 Further Related Work

3.1 Abusive Language Detection

Work on abuse detection has studied specific types of abuse using several feature-based and deep learning approaches. One of the earliest studies was by Yin et al. (2009) employing SVM to detect 'personal insult harassment' using TF-IDF values for words and sentiment-based features. Using a similar but enhanced set of features, Davidson et al. (2017) implement Logistic Regression and SVM to detect hate speech and offensive language on Twitter. Warner and Hirschberg (2012) use a template-based strategy to extract features from text and a linear-SVM to detect hate speech with a focus on anti-semitic language. Djuric et al. (2015) report an AUC of 80% using Logistic Regression with $paragraph2vec$ which outperformed standard bag-of-words approaches. Nobata et al. (2016) also use word2vec and comment2vec as one of their features to detect 'abusive language' which, in their work, encompasses hate speech, profanity and derogatory language. Wulczyn et al. (2017) implement a multilayer perceptron with word and character n-grams to detect personal attacks on Wikipedia and report an AUC of 96.5%.

In recent years, deep learning techniques have been widely adopted to detect online abuse. Pavlopoulos et al. (2017) show that RNN with GRU cells outperform the original classifier on detecting personal attacks (Wulczyn et al., 2017). Park and Fung (2017) propose a Hybrid-CNN that uses both word-level and character-level CNNs to detect hate speech on Twitter. Aken et al. (2018) utilise

Preceding Comment	Main Comment (Reply)
They were safe in turkey.	Turkey isn't even safe for (Kurdish) Turks. *Do you even watch the news?*
NH dems or Bernie bros?	*You don't really understand that term "Berniebro" do you?* Just parroting it.
Fuck the refs. Fuck the stars. Fuck Texas and every shit-kick fuck that calls it home.	*Would you like some salt with that?*
SJW's happened.	*Why the fuck did you put an apostrophe there?*
So no article, right? Thought not.	*Oh, your brain stopped functioning at that?* Well then, I'll repeat myself. The abstract is enough, the article, you can find it yourself. I'm not going to waste my time.

Table 1: Examples of unpalatable questions from our annotated dataset.

CNN, bi-directional LSTM and GRU initialised with pre-trained word embeddings and report a f1-score of 78.3% and AUC of 98.3%. Gunasekara and Nejadgholi (2018) implement a Light Gradient Boosting Machine stacking model where they combine two bidirectional-LSTM based architectures and report an AUC of 98.6%. Finally, some studies also incorporate user-level context in their models. Mishra et al. (2018) report improvements in performance by incorporating author embedding features using $node2vec$ on community graphs of Twitter users. Dadvar et al. (2013) employs both user-based features and standard content-based features to detect bullying on YouTube. There are other studies that employ user-based features to detect aggression (Chatzakou et al., 2017) and hate speech (Gao and Huang, 2017).

3.2 Linguistic Impoliteness

Long before detecting online abuse gained attention, there had been significant research on linguistic impoliteness. The most notable contribution in this field is by Culpeper (1996) who introduced his theory of impoliteness as a parallel to Brown and Levinson (1987)'s politeness theory. Impoliteness is defined as the use of strategies to attack the interlocutor's *face* – a persona that one presents in a conversation (Goffman, 1967) – and create social disruption (Culpeper, 1996). More recently, Culpeper (2010) offered conventionalized impoliteness formulae for English derived from his corpora that consisted of phone calls, 'exploitative' TV shows, army training documentaries etc. He identified candidates for impoliteness and grouped them according to structural commonalities:[2]

- *Insults*: you f*cking moron; you disgust me
- *Pointed criticisms*: this was absolutely terrible
- *Unpalatable questions*: why do you make my life impossible?
- *Dismissals*: piss off; get lost
- *Threats*: I'm gonna beat the sh*t out of you if you don't [X]

3.3 Rhetorical Questions

Rhetorical questions are defined as sentences "that have the form of a question but serve as a statement" (Anzilotti, 1982). Since unpalatable questions tend to be rhetorical in nature, we present a brief overview of the literature on rhetorical question detection in social media.

One of the first studies was on Twitter data by Li et al. (2011) where they distinguish 'qweets' – tweets that ask for some information – from other interrogative tweets including rhetorical questions. They implement SVM using a set of different handcrafted features. Bhattasali et al. (2015) use bag of n-grams to detect rhetorical questions in the Switchboard Dialogue Corpus. Their best-performing model achieved a F1-score of 0.53 by incorporating both preceding and subsequent text. Using questions from Twitter and Debate Forums, Oraby et al. (2017) implement SVM and LSTM to detect rhetorical questions, and further distinguish between sarcastic rhetorical questions and other questions. There exists other studies modeling rhetorical questions that draw inspiration from linguistic theories behind the motivations of users to post rhetorical questions (Ranganath et al., 2016, 2018). A general consensus in these studies is that rhetorical questions are hard to accurately classify due to their syntactic similarity to regular questions.

4 Data

The aim is to detect unpalatable questions in online discourse. For this, we construct a dataset using comments from Reddit and annotate them

[2] See (Culpeper, 2010) for a complete list and additional examples of each form.

for whether they contain an unpalatable question or not. We also preserve conversational context in the dataset by including the preceding comment in the discussion; therefore, our data consists of (pc_i, r_i, y_i) tuples denoting the preceding comment, reply[3] (or main comment), and the corresponding label respectively. The task is formulated as a binary classification problem where $y_i = 1$ indicates that the main comment r_i contains an unpalatable question.

We collect data from a diverse set of 15 online communities (or *subreddits*) belonging to different genres: politics, sports, hate and toxic.[4] The subreddits were carefully selected to prevent the dataset from being heavily skewed towards *not unpalatable* samples since these topics are more likely to involve opinionated and antagonistic discussions.

4.1 Question Filter

A challenge during data collection was to filter out comments that did not contain a question. We experiment with two approaches: (1) simple rule-based approach where we tokenize the comment and extract sentences that end with a '?', and (2) parsing-based approach where we first generate constituent parse trees using Stanford CoreNLP (Manning et al., 2014) and then identify questions using clause-level Penn Treebank Tags.[5]

Performance Comparison. We manually annotated a random sample of 300 Reddit comments for the presence of questions. There were a total of 81 questions out of which 74 contained a '?'. Although the parsing-based approach achieved a high precision, it missed out on several questions due to the low accuracy of the parser on noisy social media text. On the other hand, the simpler rule-based approach achieved a much higher recall and only missed out on the 7 samples that did not contain a '?'. Given the high disparity in performance, we decided to use the rule-based approach as our question filter. Although a potential data limitation, it is an acceptable design decision given that 91% of questions in our random sample were explicitly phrased using a '?'. This simple '?' heuristic has also been successfully used for identifying questions in other social media studies (Zhao and Mei, 2013; Ranganath et al., 2016; Paul et al., 2011).

	Confidence			
	0.6	**0.8**	**1.0**	*Total*
Unpalatable	879	585	453	1917
Not Unpalatable	1324	2386	5282	8992
Total	2203	2971	5735	10,909

Table 2: Distribution of confidence scores in the annotated dataset.

4.2 Crowdsourcing

We use Amazon Mechanical Turk for crowdsourcing our data annotations. The coders were shown the *main comment* and also the *preceding comment* for context. They were asked to label the main comment for whether it contained an unpalatable question or not. Each comment in our dataset is labeled by at least five different coders.

Quality Control. Since coders can sometimes be unreliable at labeling abusive content (Nobata et al., 2016), we employ three measures to ensure high quality annotations. First, we were able to provide high-quality training to our coders through the use of clear instructions that laid out detailed tips, examples, and counter examples.[6] Second, we allowed only qualified coders to contribute to the task – they were required to achieve a perfect score on a quiz which had a total of 10 questions. Third, we inserted secret test questions throughout our task to address the issue of spam responses (Kittur et al., 2008). The coders were disqualified and blocked if their accuracy on the test questions fell below our predefined threshold of 90%.

4.3 Data Description

We aggregated the five annotations by taking the majority as the final label – a data sample is considered *unpalatable* if at least 3 coders labeled it as *unpalatable*. In order to not lose useful information, we added a *confidence* dimension to the dataset which is the ratio of the number of annotations with the majority label and the total number of annotators: *confidence* $\in \{0.6, 0.8, 1.0\}$. As can be seen in Table 2, 1,917 (17.5%) comments contain an *unpalatable* question, and the remaining 82.5% of comments do not. It is interesting to note the distribution of confidence scores across the two labels. Annotators seem to be much more confident

[3]Note that we use the term *main comment* and *reply* interchangeably.

[4]See Appendix A for the complete list of subreddits.

[5]https://gist.github.com/nlothian/9240750

[6]This was done through multiple in-house annotation rounds where we improved the instructions at each step. See Appendix B for the instructions shown to the coders.

for *not unpalatable* samples: 58% of samples correspond to a confidence score of 1.0 as compared to 25% for *unpalatable* samples. Following a similar trend, 45% of comments labeled *unpalatable* have a confidence score of 0.6 as compared to only 14% for *not unpalatable* samples. This highlights the nuanced nature and complexity associated with identifying unpalatable questions.

Annotator Agreement. We compute two measures of inter-annotator reliability: (1) Cohen's Kappa, and (2) Krippendorff's alpha. Our data achieved a Kappa score of 0.82 against a random sample of 150 comments manually annotated by the authors. Out of 150 comments, there were a total of 8 instances of disagreement – 7 out of which had a confidence score of 0.6. Next, we compute Krippendorff's α which is used when there are multiple coders annotating overlapping but different sets of comments (Krippendorff, 2004). Our data achieved an $\alpha = 0.39$ which is in-line with other abuse detection work that used crowdsourcing (Wulczyn et al., 2017; Cheng et al., 2015).

4.4 Comparison with Existing Datasets

As previously discussed, most datasets used for abuse detection contain annotations only for standalone comments in isolation. This is problematic since offensiveness can highly depend on the context. Castelle (2018) shows how their learning models failed (F1 = 0.3) on a StackOverflow dataset that required contextual enrichment to determine the offensiveness of a comment – a majority of comments, that were originally flagged as offensive, were not considered offensive by their coders. This is because the dataset lacked interactional context which was available to StackOverflow users when they originally flagged it as offensive.

We are aware of the following existing datasets that include contextual information:[7]

- Karan and Šnajder (2019) published a large dataset of 400k comments from Wikipedia including complete discussion threads. However, a major limitation of their data is that the labels are generated automatically using an existing toxicity classifier.[8] This implies that their labels would not be accurate for comments where the original toxicity classifier itself fails. In comparison, we perform manual annotation where our coders explicitly consider interactional context.

- Liu et al. (2018) published a dataset of 30,987 comments from Instagram annotated for *hostility*. The coders were shown an Instagram post and all comments in the thread. However, their data collection is biased (intentionally) towards teenagers and is filtered by certain keywords, eg: profanities, emojis. In comparison, our dataset involves a random sample of comments from a diverse set of subreddits without the use of any keyword-filtering.

- Gao and Huang (2017) released a dataset of 10 complete discussion threads from Fox News. The data includes additional contextual information in the form of user screen name, other comments in the thread, and title of the news article. However, their dataset is much smaller with only 1,528 comments and includes only two annotations per comment. In comparison, we use at least five annotations for each of the 10,909 comments in our dataset.

5 Methodology

In this section, we introduce our methodology for detecting unpalatable questions.

5.1 Traditional Machine Learning

We implement Logistic Regression[9] using a diverse set of features:

- **N-grams**: We use TF-IDF values for word unigrams, bigrams, and trigrams. We also utilise character trigrams, 4-grams, and 5-grams.

- **Embeddings**: We experiment with several pre-trained 300-dimensional embeddings: word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), and Fasttext (Bojanowski et al., 2017). In addition, we trained a word2vec model[10] from scratch on Reddit using Gensim (Řehůřek and Sojka, 2010).

- **Writing Style**: This category captures writing style of the comment and includes the following features: total number of words,

[7] Additionally, Zhang et al. (2018) released a dataset to *pre-emptively* detect toxic comments given all preceding comments in the discussion focusing on "personal attacks."

[8] http://www.perspectiveapi.com

[9] Note that additional experiments with SVM yielded similar results (not shown here).

[10] Specifically, we train the continuous bag-of-words (CBOW) architecture on all of Reddit data from 2016.

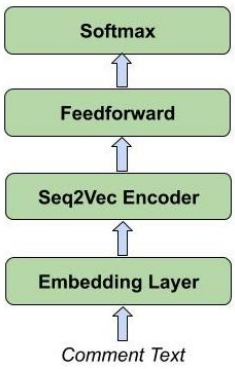


Figure 2: The skeletal architecture for our deep learning models.

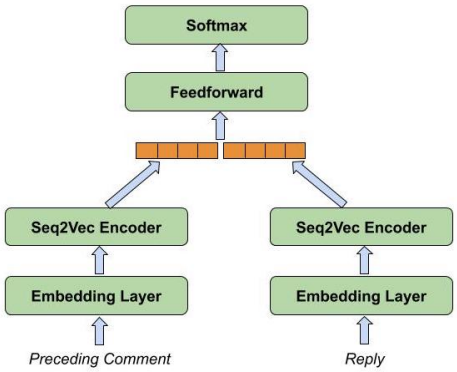


Figure 3: The skeletal architecture for our deep learning models that incorporate interactional context.

capital words, question marks, exclamation marks and second person pronouns.

- **Lexicon-based**: This category includes three features computed using pre-defined lexicons:
  - We compute the number of non-English words by comparing against NLTK words (Loper and Bird, 2002) and Enchant's dictionary.[11]
  - We compute the number of toxic words using a lexicon compiled from different sources: list of bad words released by Google[12] and lexicons released by other studies (Chandrasekharan et al., 2017; Davidson et al., 2017).
  - Empath (Fast et al., 2016) provides a set of 200 built-in validated categories for analysing text. We hand-picked a set of 15 relevant categories, for example: aggression, disgust, hate, shame etc.
- **Sentiment**: We use the *positive*, *negative*, and *neutral* sentiment scores returned by VADER, a rule-based sentiment analyser built for social media text (Hutto and Gilbert, 2014).

To build feature vectors, we experiment with the features in isolation as well as several combinations of these feature categories. The feature vectors along with the corresponding labels y_i are then fed to the learning algorithm, which is implemented using scikit-learn (Pedregosa et al., 2011).

5.2 Deep Learning

Deep learning models have been successfully used in many abuse detection studies (Pavlopoulos et al., 2017; Park and Fung, 2017; Aken et al., 2018). In this work, we implement a number of deep learning models – both CNN and RNN-based – using the architecture shown in Figure 2. We use pre-trained GloVe embeddings for the embedding layer.[13] An encoder is responsible for condensing a sequence of word vectors to a single vector. We experiment with a number of neural networks for the encoder: CNN, LSTM, Bidirectional LSTM, and Stacked Bidirectional LSTM.

ELMo. For the embedding layer, we also experiment with deep contextualized ELMo (Peters et al., 2018) representations[14] as an alternative to using GloVe embeddings. The encoder layer here can be either a CNN, LSTM, Bi-LSTM, or Stacked Bi-LSTM.

Dense Hybrid. We also implement deep learning models that utilise the various hand-engineered features discussed in Section 5.1. For this, we compute a 'dense' feature vector using those feature categories, and concatenate it with the neural encoder's output. This combined vector is then fed to a fully-connected feedforward neural network followed by a softmax layer.

Implementation details. All models are implemented using AllenNLP (Gardner et al., 2017), an open-source deep learning library for NLP. The training objective is weighted cross entropy loss, and Adam optimizer (Kingma and Ba, 2014) is used for learning network weights. Additionally, early stopping is implemented to terminate training of the neural network once the loss stops improving on a set-aside validation set.

[11] https://github.com/rfk/pyenchant

[12] https://code.google.com/archive/p/badwordslist/downloads

[13] We picked GloVe since it showed the best performance in our traditional machine learning experiments.

[14] 'Original 5.5B' model from https://allennlp.org/elmo.

Text Input	Model	F1	AUROC	W-F1	Prec.	Rec.	AUPRC
Reply Text	Dense CNN + ELMo	**0.532**	0.818	0.815	0.459	0.636	0.54
Reply Text + Comment Text	CNN + ELMo	0.52	0.814	0.806	0.444	0.636	0.527
Reply Text + Comment Text	NLI-CNN + ELMo	0.507	0.810	0.805	0.443	0.602	0.515
Reply Text + Comment Text	Dense CNN (GloVe)	0.5	0.8	0.825	0.507	0.496	0.502
Question Text Only	Dense Bi-LSTM + ELMo	0.5	0.809	0.778	0.394	0.692	0.517
Reply Text	word (1, 3)	0.443	0.782	0.823	0.549	0.373	0.481
Reply Text	word (1, 3) + Sentiment + Writing-Style + Lexicon	0.429	0.785	0.825	0.6	0.335	0.491

Table 3: Classification results for learning models on the *All-Data* scenario: *confidence* $\in \{0.6, 0.8, 1.0\}$

Text Input	Model	F1	AUROC	W-F1	Prec.	Rec.	AUPRC
Question Text Only	Dense CNN + ELMo	**0.686**	0.937	0.95	0.672	0.704	0.739
Reply Text + Comment Text	NLI-CNN + ELMo	0.674	0.937	0.95	0.74	0.625	0.721
Reply Text	CNN + ELMo	0.671	0.936	0.949	0.688	0.662	0.727
Question Text Only	Stacked Bi-LSTM + ELMo	0.638	0.931	0.938	0.569	0.73	0.718
Reply Text + Comment Text	Dense CNN (GloVe)	0.63	0.936	0.945	0.754	0.56	0.702
Question Text Only	Writing Style + Lexicon + Sentiment + GloVe	0.618	0.906	0.942	0.671	0.574	0.648
Question Text Only	char (3, 5) + Sentiment + Writing-Style + Lexicon	0.614	0.916	0.944	0.753	0.521	0.672

Table 4: Classification results for learning models on the *High-Agreement-Data* scenario: *confidence* $= 1.0$

5.3 Incorporating Conversational Context

Since humans can better comprehend a comment with reference to its context, we wanted to investigate the benefits of incorporating conversational context in the learning models. For traditional machine learning models, we concatenate the feature vectors for the preceding comment pc_i and main comment r_i which is then fed to the learning algorithm. For deep learning models, we first vectorize the preceding comment pc_i and the main comment r_i using the same encoder pipeline. The two vectors are then concatenated and fed to a feedforward neural network (Figure 3). In the LSTM-based models, the final hidden states of the pc_i and r_i pipeline are concatenated. For CNN, the output of the max pooling layers of the pc_i and r_i pipeline are concatenated.

In addition to simple concatenation, we experiment with additional heuristics to model context inspired from the task of Natural Language Inference (NLI). Specifically, we used a CNN encoder to vectorize the context pc_i and main comment r_i. They are then combined using three-heuristics: (1) concatenation, (2) element-wise product, and (3) element-wise difference (Mou et al., 2016).

6 Experiments and Results

We conduct experiments across two dimensions:

1. **Confidence Score:** We hypothesize that the models would exhibit better performance on data samples which were easier for the coders to annotate. To test this, we experiment with two scenarios:
   - *All-Data*: we use the complete dataset.
   - *High-Agreement-Data*: we use data samples corresponding to *confidence* $= 1.0$.
2. **Text Input:** To investigate the benefits of including contextual information, we experiment with three input scenarios:
   - *Question Text Only*: we only provide the question text as input.
   - *Reply Text*: we provide the full text of the main comment as input.
   - *Reply Text + Comment Text*: In addition to the main comment, we also provide the preceding comment text as input.

We evaluate our computational models on several classification metrics: precision, recall, F1-score, and Area under Receiver Operating Characteristic curve (AUROC) and Precision-Recall curve (AUPRC). All reported values are averaged over stratified five-fold cross-validation runs. The empirical results on *All-Data* and *High-Agreement-Data* are presented in Table 3 and Table 4 respectively.[15]

[15] We only display the top-performing traditional and deep learning models here. The complete list of all results is available in Appendix C.

7 Discussion

Among traditional learning algorithms, a combination of simple word unigrams, bigrams, and trigrams achieves the best F1-score of 0.44. Adding other hand-engineered features to $word(1,3)$ results in a better precision and AUPRC. As expected, deep learning models outperform traditional machine learning algorithms for both *All-Data* and *High-Agreement-Data* scenarios. In particular, CNN models perform much better than LSTM models. This is evident from Tables 3 and 4 where the best-CNN model outperforms the best-LSTM model by a 3-point and 5-point increase in F1-score respectively. We observe improvements with using contextualized ELMo embeddings as opposed to static GloVe embeddings for both scenarios. Moreover, the addition of dense hand-engineered feature vector further improves the F1-score to 0.532. Finally, our hypothesis, that it would be easier for the models to classify if it was easier for the humans, holds true in that there is a considerable improvement in the F1-score (15 points) for *High-Agreement-Data*.

Despite the performance gains observed with using more sophisticated deep learning models, the performance is still poor to be used for any practical applications. This is not surprising given how linguistically nuanced our dataset is and the complexity associated with abusive language detection on noisy social media text (Nobata et al., 2016). Specifically, learning models struggle to deal with *implicit* abuse – language which does not immediately convey abuse (Waseem et al., 2017). Aken et al. (2018) find that their toxicity classifier fails on data where there were instances of sarcasm, toxicity without employing swear words, and rhetorical questions. We qualitatively examined a random sample of hundred *unpalatable* samples from our dataset, and found that 65% do not contain swear words and 20% involve sarcasm. Similarly, from a random sample of hundred mis-classified *unpalatable* samples, 72% do not contain swear words and 30% involve sarcasm. Moreover, since unpalatable questions are rhetorical in nature, it is not surprising that learning models performed relatively poorly on the task.

Context. Our assumption was that models would benefit from conversational context since humans find it easier to determine the offensiveness of a comment when provided with some context. It is, however, evident from our empirical results that incorporating context through providing the preceding comment to the model did not improve performance for both traditional machine learning and deep learning models. This finding is consistent with other studies that attempt to incorporate interactional context into their models (Karan and Šnajder, 2019; Lee et al., 2018). We believe that effectively incorporating deeper context, as opposed to just the preceding comment, using more sophisticated methods such as hierarchical neural networks might help improve performance.

Evaluation Metrics. Mishra et al. (2019) observe a problematic trend with several abuse detection studies using AUROC for evaluation. This is not ideal since ROC plots can be deceptive when dealing with imbalanced classification scenarios (Saito and Rehmsmeier, 2015). Since most abuse detection datasets tend to be heavily skewed towards non-abusive samples, this can lead to misleadingly optimistic values for AUROC (also observed in Tables 3 and 4). A better alternative is to report AUPRC which is more robust to imbalanced data since it evaluates the fraction of true positives among positive predictions at different thresholds (Saito and Rehmsmeier, 2015).

8 Conclusion

In this work, we addressed an important gap in the abuse detection literature by introducing a novel task of detecting unpalatable questions. We also released a context-rich dataset for the task and implemented a number of learning models to automatically detect unpalatable questions. Our results show that it is difficult to model subtle abuse due to the language being nuanced and context-sensitive. This calls for advancements in natural language understanding methods that can identify such implicit signals and take pragmatic context into account. We hope that future research would explore other forms of abuse and draw inspiration from related fields such as linguistic impoliteness. Detecting abuse – both overt and subtle – on the Internet would help enhance user's experience online and facilitate civil and productive discussions.

References

Betty Van Aken, Julian Risch, Ralf Krestel, and Alexander Löser. 2018. Challenges for toxic comment classification: An in-depth error analysis. In

Proceedings of the 2nd Workshop on Abusive Language Online (ALW2), pages 33–42, Brussels, Belgium. Association for Computational Linguistics.

Gloria Italiano Anzilotti. 1982. The rhetorical question as an indirect speech device in english and italian. *The Canadian Modern Language Review*, 38(2):290–302.

Shohini Bhattasali, Jeremy Cytryn, Elana Feldman, and Joonsuk Park. 2015. Automatic identification of rhetorical questions. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 743–749, Beijing, China. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Derek Bousfield. 2007. Impoliteness, preference organization and conducivity. *Multilingua – Journal of Cross-Cultural and Interlanguage Communication*, 26(1):1–33.

Penelope Brown and Stephen Levinson. 1987. *Politeness: Some universals in language usage*. Cambridge University Press, New York, NY, US.

Andrew Caines, Sergio Pastrana, Alice Hutchings, and Paula Buttery. 2018. Aggressive language in an online hacking forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 66–74, Brussels, Belgium. Association for Computational Linguistics.

Michael Castelle. 2018. The linguistic ideologies of deep abusive language classification. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 160–170, Brussels, Belgium. Association for Computational Linguistics.

Eshwar Chandrasekharan, Umashanthi Pavalanathan, Anirudh Srinivasan, Adam Glynn, Jacob Eisenstein, and Eric Gilbert. 2017. You can't stay here: The efficacy of reddit's 2015 ban examined through hate speech. *Proc. ACM Hum.-Comput. Interact.*, 1(CSCW):31:1–31:22.

Despoina Chatzakou, Nicolas Kourtellis, Jeremy Blackburn, Emiliano De Cristofaro, Gianluca Stringhini, and Athena Vakali. 2017. Mean birds: Detecting aggression and bullying on twitter. In *Proceedings of the 2017 ACM on Web Science Conference*, WebSci '17, pages 13–22, New York, NY, USA. ACM.

Justin Cheng, Cristian Danescu-Niculescu-Mizil, and Jure Leskovec. 2015. Antisocial behavior in online discussion communities. In *Proceedings of ICWSM*.

Jonathan Culpeper. 1996. Towards an anatomy of impoliteness. *Journal of Pragmatics*, 25(3):349–367.

Jonathan Culpeper. 2010. Conventionalised impoliteness formulae. *Journal of Pragmatics*, 42(12):3232 – 3245.

Maral Dadvar, Dolf Trieschnigg, Roeland Ordelman, and Franciska de Jong. 2013. Improving cyberbullying detection with user context. In *Advances in Information Retrieval*, pages 693–696, Berlin, Heidelberg. Springer Berlin Heidelberg.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the 11th International AAAI Conference on Web and Social Media*, ICWSM '17, pages 512–515.

Nemanja Djuric, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, and Narayan Bhamidipati. 2015. Hate speech detection with comment embeddings. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, pages 29–30, New York, NY, USA. ACM.

Maeve Duggan. 2017. *Online Harassment 2017*. Pew Research Center.

Ethan Fast, Binbin Chen, and Michael S. Bernstein. 2016. Empath: Understanding topic signals in large-scale text. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 4647–4657, New York, NY, USA. ACM.

Antigoni Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018. Large scale crowdsourcing and characterization of twitter abusive behavior. In *International AAAI Conference on Web and Social Media*.

Lei Gao and Ruihong Huang. 2017. Detecting online hate speech using context aware models. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 260–266, Varna, Bulgaria. INCOMA Ltd.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. Allennlp: A deep semantic natural language processing platform.

E. Goffman. 1967. *Interaction Ritual: Essays in Face-to-face Behavior*. Aldine Publishing Company.

Jennifer Golbeck, Zahra Ashktorab, Rashad O. Banjo, Alexandra Berlinger, Siddharth Bhagwan, Cody Buntain, Paul Cheakalos, Alicia A. Geller, Quint Gergory, Rajesh Kumar Gnanasekaran, Raja Rajan Gunasekaran, Kelly M. Hoffman, Jenny Hottle, Vichita Jienjitlert, Shivika Khare, Ryan Lau, Marianna J. Martindale, Shalmali Naik, Heather L. Nixon, Piyush Ramachandran, Kristine M. Rogers,

Lisa Rogers, Meghna Sardana Sarin, Gaurav Shahane, Jayanee Thanki, Priyanka Vengataraman, Zijian Wan, and Derek Michael Wu. 2017. A large labeled corpus for online harassment research. In *Proceedings of the 2017 ACM on Web Science Conference*, WebSci '17, pages 229–233, New York, NY, USA. ACM.

Isuru Gunasekara and Isar Nejadgholi. 2018. A review of standard text classification practices for multi-label toxicity identification of online content. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 21–25, Brussels, Belgium. Association for Computational Linguistics.

Clayton J. Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *ICWSM*.

David Jurgens, Libby Hemphill, and Eshwar Chandrasekharan. 2019. A just and comprehensive strategy for using NLP to address online abuse. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3658–3666, Florence, Italy. Association for Computational Linguistics.

Mladen Karan and Jan Šnajder. 2019. Preemptive toxic language detection in Wikipedia comments using thread-level context. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 129–134, Florence, Italy. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Aniket Kittur, Ed H. Chi, and Bongwon Suh. 2008. Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 453–456, New York, NY, USA. ACM.

Klaus Krippendorff. 2004. Reliability in content analysis. *Human Communication Research*, 30(3):411–433.

Younghun Lee, Seunghyun Yoon, and Kyomin Jung. 2018. Comparative studies of detecting abusive language on twitter. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 101–106, Brussels, Belgium. Association for Computational Linguistics.

Baichuan Li, Xiance Si, Michael R. Lyu, Irwin King, and Edward Y. Chang. 2011. Question identification on twitter. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 2477–2480, New York, NY, USA. ACM.

Ping Liu, Joshua Guberman, Libby Hemphill, and Aron Culotta. 2018. Forecasting the presence and intensity of hostility on instagram using linguistic and social features. In *International AAAI Conference on Web and Social Media*.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. In *Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, USA. Curran Associates Inc.

Pushkar Mishra, Marco Del Tredici, Helen Yannakoudakis, and Ekaterina Shutova. 2018. Author profiling for abuse detection. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1088–1098, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Pushkar Mishra, Helen Yannakoudakis, and Ekaterina Shutova. 2019. Tackling online abuse: A survey of automated abuse detection methods.

Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. 2016. Natural language inference by tree-based convolution and heuristic matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 130–136, Berlin, Germany. Association for Computational Linguistics.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pages 145–153, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Shereen Oraby, Vrindavan Harrison, Amita Misra, Ellen Riloff, and Marilyn Walker. 2017. Are you serious?: Rhetorical questions and sarcasm in social media dialog. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 310–319, Saarbrücken, Germany. Association for Computational Linguistics.

Ji Ho Park and Pascale Fung. 2017. One-step and two-step classification for abusive language detection on twitter. In *Proceedings of the First Workshop on Abusive Language Online*, pages 41–45, Vancouver, BC, Canada. Association for Computational Linguistics.

Sharoda Paul, Lichan Hong, and Ed Chi. 2011. Is twitter a good place for asking questions? a characterization study. In *International AAAI Conference on Web and Social Media*.

John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. 2017. Deeper attention to abusive user content moderation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1125–1135, Copenhagen, Denmark. Association for Computational Linguistics.

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

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. 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 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Suhas Ranganath, Xia Hu, Jiliang Tang, Suhang Wang, and Huan Liu. 2016. Identifying rhetorical questions in social media. In *International AAAI Conference on Web and Social Media*.

Suhas Ranganath, Xia Hu, Jiliang Tang, Suhang Wang, and Huan Liu. 2018. Understanding and identifying rhetorical questions in social media. *ACM Trans. Intell. Syst. Technol.*, 9(2):17:1–17:22.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA. http://is.muni.cz/publication/884893/en.

Takaya Saito and Marc Rehmsmeier. 2015. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3).

Niloofar Safi Samghabadi, Suraj Maharjan, Alan Sprague, Raquel Diaz-Sprague, and Thamar Solorio. 2017. Detecting nastiness in social media. In *Proceedings of the First Workshop on Abusive Language Online*, pages 63–72, Vancouver, BC, Canada. Association for Computational Linguistics.

William Warner and Julia Hirschberg. 2012. Detecting hate speech on the world wide web. In *Proceedings of the Second Workshop on Language in Social Media*, pages 19–26, Montréal, Canada. Association for Computational Linguistics.

Zeerak Waseem, Thomas Davidson, Dana Warmsley, and Ingmar Weber. 2017. Understanding abuse: A typology of abusive language detection subtasks. In *Proceedings of the First Workshop on Abusive Language Online*, pages 78–84, Vancouver, BC, Canada. Association for Computational Linguistics.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL Student Research Workshop*, pages 88–93, San Diego, California. Association for Computational Linguistics.

Agus Wijayanto, Aryati Prasetyarini, and Mauly Halwat Hikmat. 2017. Impoliteness in efl: Foreign language learners' complaining behaviors across social distance and status levels. *SAGE Open*, 7(3).

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 1391–1399, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Dawei Yin, Brian D. Davison, Zhenzhen Xue, Liangjie Hong, April Kontostathis, and Lynne Edwards. 2009. Detection of harassment on web 2.0.

Justine Zhang, Jonathan Chang, Cristian Danescu-Niculescu-Mizil, Lucas Dixon, Yiqing Hua, Dario Taraborelli, and Nithum Thain. 2018. Conversations gone awry: Detecting early signs of conversational failure. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1350–1361, Melbourne, Australia. Association for Computational Linguistics.

Zhe Zhao and Qiaozhu Mei. 2013. Questions about questions: An empirical analysis of information needs on twitter. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, pages 1545–1556, New York, NY, USA. ACM.

Brett Zimmerman. 2005. *Catalogue of Rhetorical and Other Literary Terms in Poe's Works*, pages 107–325. McGill-Queen's University Press.

A Data Collection

We collect data from a diverse set of 15 Reddit communities (or *subreddits*) belonging to different genres:

- **Politics**:

  - r/The_Donald
  - r/politics
  - r/PoliticalDiscussion
  - r/Conservative

- **Sports**:
  - r/nfl
  - r/sports
  - r/nba
  - r/hockey

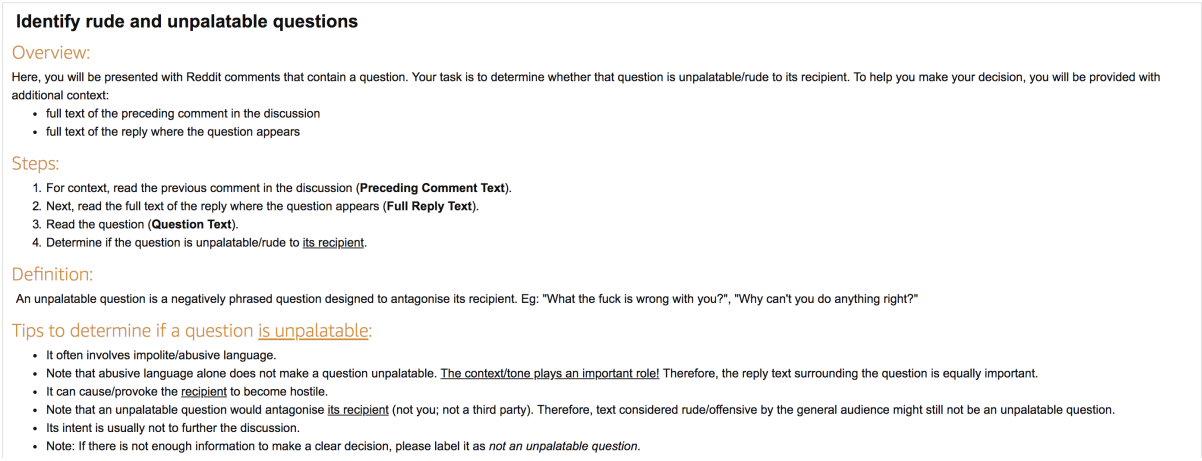
- **Hate and Toxic**:
  - r/cringepics
  - r/cringe
  - r/4chan
  - r/CringeAnarchy
  - r/KotakuInAction
  - r/ImGoingToHellForThis
  - r/TumblrInAction

B Crowdsourcing Instructions

The instructions provided to Amazon Mechanical Turk coders are shown in Figure 4 and Figure 5.

C Results

The complete list of results for the *All-Data* scenario are shown in Table 5 (deep learning models) and Table 7 (traditional machine learning models). Next, the complete list of results for the *High-Agreement-Data* scenario are shown in Table 6 (deep learning models) and Table 8 (traditional machine learning models).



Identify rude and unpalatable questions

Overview:

Here, you will be presented with Reddit comments that contain a question. Your task is to determine whether that question is unpalatable/rude to its recipient. To help you make your decision, you will be provided with additional context:

- full text of the preceding comment in the discussion
- full text of the reply where the question appears

Steps:

1. For context, read the previous comment in the discussion (**Preceding Comment Text**).
2. Next, read the full text of the reply where the question appears (**Full Reply Text**).
3. Read the question (**Question Text**).
4. Determine if the question is unpalatable/rude to its recipient.

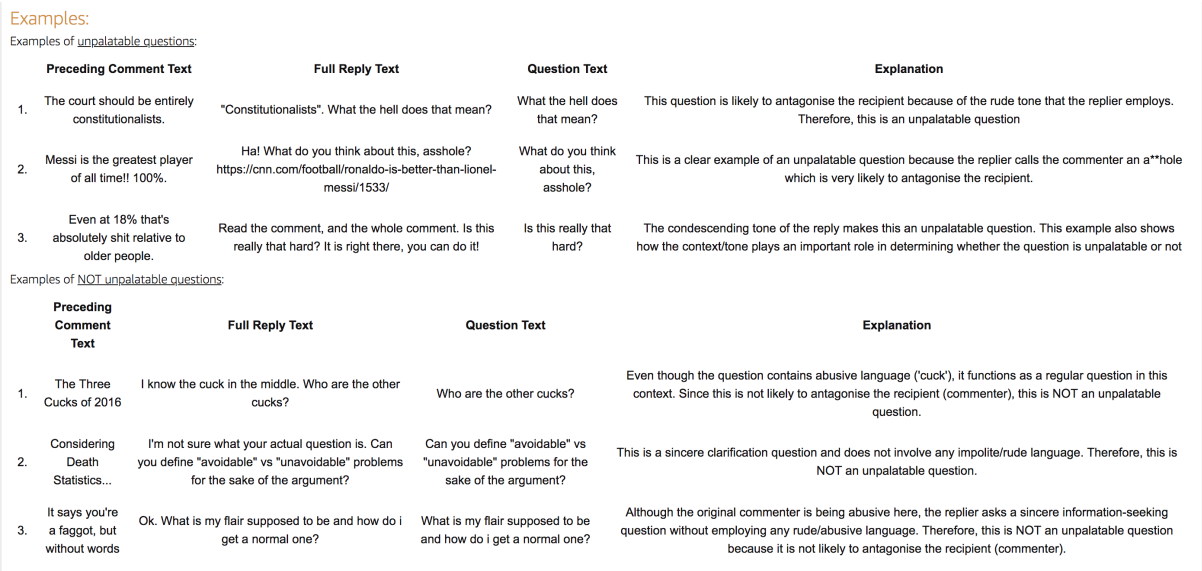
Definition:

An unpalatable question is a negatively phrased question designed to antagonise its recipient. Eg: "What the fuck is wrong with you?", "Why can't you do anything right?"

Tips to determine if a question is unpalatable:

- It often involves impolite/abusive language.
- Note that abusive language alone does not make a question unpalatable. The context/tone plays an important role! Therefore, the reply text surrounding the question is equally important.
- It can cause/provoke the recipient to become hostile.
- Note that an unpalatable question would antagonise its recipient (not you; not a third party). Therefore, text considered rude/offensive by the general audience might still not be an unpalatable question.
- Its intent is usually not to further the discussion.
- Note: If there is not enough information to make a clear decision, please label it as *not an unpalatable question*.

Figure 4: Instructions for the crowdsourcing task as seen by Mechanical Turk Workers.



Examples:

Examples of unpalatable questions:

	Preceding Comment Text	Full Reply Text	Question Text	Explanation
1.	The court should be entirely constitutionalists.	"Constitutionalists". What the hell does that mean?	What the hell does that mean?	This question is likely to antagonise the recipient because of the rude tone that the replier employs. Therefore, this is an unpalatable question
2.	Messi is the greatest player of all time!! 100%.	Ha! What do you think about this, asshole? https://cnn.com/football/ronaldo-is-better-than-lionel-messi/1533/	What do you think about this, asshole?	This is a clear example of an unpalatable question because the replier calls the commenter an a**hole which is very likely to antagonise the recipient.
3.	Even at 18% that's absolutely shit relative to older people.	Read the comment, and the whole comment. Is this really that hard? It is right there, you can do it!	Is this really that hard?	The condescending tone of the reply makes this an unpalatable question. This example also shows how the context/tone plays an important role in determining whether the question is unpalatable or not

Examples of NOT unpalatable questions:

	Preceding Comment Text	Full Reply Text	Question Text	Explanation
1.	The Three Cucks of 2016	I know the cuck in the middle. Who are the other cucks?	Who are the other cucks?	Even though the question contains abusive language ('cuck'), it functions as a regular question in this context. Since this is not likely to antagonise the recipient (commenter), this is NOT an unpalatable question.
2.	Considering Death Statistics...	I'm not sure what your actual question is. Can you define "avoidable" vs "unavoidable" problems for the sake of the argument?	Can you define "avoidable" vs "unavoidable" problems for the sake of the argument?	This is a sincere clarification question and does not involve any impolite/rude language. Therefore, this is NOT an unpalatable question.
3.	It says you're a faggot, but without words	Ok. What is my flair supposed to be and how do i get a normal one?	What is my flair supposed to be and how do i get a normal one?	Although the original commenter is being abusive here, the replier asks a sincere information-seeking question without employing any rude/abusive language. Therefore, this is NOT an unpalatable question because it is not likely to antagonise the recipient (commenter).

Figure 5: Additional examples for the crowdsourcing task as seen by Mechanical Turk Workers.

Text Input	Model	F1-score	AUROC	Weighted F1	Precision	Recall	Accuracy	AUPRC
Reply Text	Dense CNN (ELMo)	**0.532**	0.818	0.815	0.459	0.636	0.803	0.54
Reply Text + Comment Text	CNN (ELMo)	0.52	0.814	0.806	0.444	0.636	0.792	0.527
Reply Text + Comment Text	Dense CNN (ELMo)	0.516	0.815	0.816	0.476	0.578	0.809	0.531
Question Text Only	CNN (ELMo)	0.514	0.811	0.802	0.434	0.643	0.787	0.532
Reply Text	CNN (ELMo)	0.509	0.814	0.787	0.406	0.688	0.766	0.537
Reply Text + Comment Text	NLI-CNN (ELMo)	0.507	0.810	0.805	0.443	0.602	0.792	0.515
Question Text Only	Dense CNN (ELMo)	0.502	0.813	0.78	0.401	0.689	0.758	0.54
Question Text Only	Dense Bi-LSTM (ELMo)	0.5	0.809	0.778	0.394	0.692	0.755	0.517
Reply Text + Comment Text	Dense CNN (GloVe)	0.5	0.8	0.825	0.507	0.496	0.826	0.502
Question Text Only	Bi-LSTM (ELMo)	0.5	0.804	0.778	0.391	0.697	0.755	0.513
Reply Text	Stacked Bi-LSTM (ELMo)	0.497	0.803	0.782	0.396	0.67	0.76	0.501
Reply Text + Comment Text	Dense Bi-LSTM (ELMo)	0.494	0.802	0.777	0.391	0.683	0.754	0.489
Reply Text + Comment Text	CNN (GloVe)	0.491	0.802	0.826	0.517	0.471	0.829	0.51
Reply Text	Dense LSTM (ELMo)	0.49	0.794	0.768	0.378	0.702	0.742	0.489
Question Text Only	LSTM (ELMo)	0.489	0.802	0.767	0.378	0.7	0.742	0.509
Reply Text	Bi-LSTM (ELMo)	0.489	0.799	0.765	0.376	0.707	0.739	0.494
Reply Text	Dense Stacked Bi-LSTM (ELMo)	0.487	0.797	0.773	0.386	0.673	0.75	0.49
Reply Text + Comment Text	LSTM (ELMo)	0.487	0.791	0.768	0.381	0.687	0.744	0.453
Reply Text + Comment Text	Dense Stacked Bi-LSTM (ELMo)	0.484	0.791	0.783	0.407	0.626	0.766	0.468
Reply Text	Dense Bi-LSTM (ELMo)	0.484	0.803	0.771	0.398	0.661	0.75	0.503
Question Text Only	Dense Stacked Bi-LSTM (ELMo)	0.482	0.804	0.754	0.372	0.719	0.726	0.516
Reply Text + Comment Text	Bi-LSTM (ELMo)	0.481	0.801	0.757	0.366	0.714	0.729	0.493
Reply Text	LSTM (ELMo)	0.481	0.793	0.748	0.355	0.747	0.716	0.484
Question Text Only	CNN (GloVe)	0.48	0.782	0.809	0.451	0.517	0.804	0.504
Reply Text + Comment Text	Stacked Bi-LSTM (ELMo)	0.479	0.793	0.75	0.36	0.726	0.721	0.466
Question Text Only	Stacked Bi-LSTM (ELMo)	0.478	0.8	0.744	0.352	0.746	0.712	0.507
Question Text Only	Dense LSTM (ELMo)	0.477	0.804	0.74	0.349	0.76	0.707	0.514
Reply Text	CNN (GloVe)	0.473	0.792	0.817	0.488	0.466	0.818	0.498
Question Text Only	Dense CNN (GloVe)	0.472	0.765	0.788	0.416	0.562	0.775	0.479
Reply Text + Comment Text	NLI-CNN (GloVe)	0.471	0.811	0.829	0.573	0.407	0.841	0.516
Reply Text + Comment Text	Dense LSTM (ELMo)	0.468	0.783	0.748	0.357	0.698	0.72	0.448
Reply Text + Comment Text	Stacked Bi-LSTM (GloVe)	0.461	0.767	0.795	0.422	0.524	0.787	0.443
Reply Text	Dense Bi-LSTM (GloVe)	0.461	0.755	0.805	0.446	0.485	0.801	0.451
Reply Text	Dense CNN (GloVe)	0.459	0.787	0.802	0.457	0.49	0.799	0.489
Reply Text	Dense LSTM (GloVe)	0.458	0.742	0.8	0.435	0.492	0.795	0.436
Reply Text	Dense Stacked Bi-LSTM (GloVe)	0.456	0.753	0.806	0.45	0.468	0.805	0.452
Reply Text + Comment Text	Dense LSTM (GloVe)	0.455	0.752	0.796	0.419	0.504	0.789	0.423
Reply Text + Comment Text	Dense Bi-LSTM (GloVe)	0.455	0.758	0.802	0.446	0.48	0.798	0.448
Question Text Only	LSTM (GloVe)	0.452	0.736	0.776	0.394	0.555	0.761	0.435
Reply Text + Comment Text	Bi-LSTM (GloVe)	0.451	0.762	0.805	0.457	0.459	0.804	0.446
Reply Text + Comment Text	Dense Stacked Bi-LSTM (GloVe)	0.446	0.75	0.81	0.478	0.423	0.814	0.444
Question Text Only	Dense LSTM (GloVe)	0.446	0.735	0.773	0.39	0.554	0.758	0.448
Question Text Only	Dense Bi-LSTM (GloVe)	0.446	0.735	0.798	0.427	0.473	0.794	0.452
Question Text Only	Dense Stacked Bi-LSTM (GloVe)	0.442	0.738	0.746	0.352	0.617	0.72	0.446
Question Text Only	Stacked Bi-LSTM (GloVe)	0.439	0.733	0.768	0.376	0.551	0.751	0.447
Reply Text + Comment Text	LSTM (GloVe)	0.438	0.742	0.793	0.451	0.462	0.79	0.422
Question Text Only	Bi-LSTM (GloVe)	0.437	0.738	0.765	0.368	0.557	0.746	0.438
Reply Text	Stacked Bi-LSTM (GloVe)	0.435	0.749	0.79	0.438	0.471	0.785	0.435
Reply Text	LSTM (GloVe)	0.434	0.74	0.767	0.383	0.556	0.751	0.426
Reply Text	Bi-LSTM (GloVe)	0.421	0.731	0.75	0.353	0.55	0.729	0.404

Table 5: Classification results for deep learning models on the complete dataset: *confidence* $\in \{0.6, 0.8, 1.0\}$

Text Input	Model	F1-score	AUROC	Weighted F1	Precision	Recall	Accuracy	AUPRC
Question Text Only	Dense CNN (ELMo)	**0.686**	0.937	0.95	0.672	0.704	0.949	0.739
Reply Text + Comment Text	NLI-CNN (ELMo)	0.674	0.937	0.95	0.74	0.625	0.953	0.721
Reply Text	CNN (ELMo)	0.671	0.936	0.949	0.688	0.662	0.949	0.727
Question Text Only	CNN (ELMo)	0.669	0.937	0.946	0.643	0.702	0.945	0.734
Reply Text	Dense CNN (ELMo)	0.668	0.939	0.949	0.709	0.634	0.95	0.736
Reply Text + Comment Text	Dense CNN (ELMo)	0.667	0.941	0.949	0.73	0.618	0.951	0.735
Reply Text + Comment Text	CNN (ELMo)	0.665	0.941	0.948	0.695	0.655	0.949	0.737
Question Text Only	Stacked Bi-LSTM (ELMo)	0.638	0.931	0.938	0.569	0.73	0.934	0.718
Reply Text + Comment Text	Dense CNN (GloVe)	0.63	0.936	0.945	0.754	0.56	0.949	0.702
Question Text Only	Dense Bi-LSTM (ELMo)	0.628	0.931	0.937	0.568	0.715	0.934	0.714
Reply Text	CNN (GloVe)	0.626	0.932	0.945	0.754	0.538	0.949	0.711
Question Text Only	Dense CNN (GloVe)	0.625	0.926	0.944	0.705	0.572	0.946	0.695
Question Text Only	Bi-LSTM (ELMo)	0.623	0.931	0.934	0.543	0.735	0.929	0.711
Reply Text	Dense Stacked Bi-LSTM (ELMo)	0.61	0.915	0.934	0.553	0.687	0.931	0.651
Question Text Only	CNN (GloVe)	0.609	0.92	0.94	0.673	0.569	0.942	0.692
Reply Text + Comment Text	CNN (GloVe)	0.605	0.936	0.944	0.777	0.499	0.949	0.708
Reply Text + Comment Text	Bi-LSTM (GloVe)	0.603	0.9	0.942	0.712	0.525	0.946	0.644
Reply Text + Comment Text	Dense Bi-LSTM (ELMo)	0.603	0.914	0.934	0.565	0.653	0.932	0.638
Reply Text	Dense CNN (GloVe)	0.603	0.934	0.94	0.682	0.549	0.943	0.703
Reply Text + Comment Text	Dense LSTM (ELMo)	0.602	0.9	0.934	0.553	0.664	0.931	0.586
Reply Text + Comment Text	NLI-CNN (GloVe)	0.6	0.937	0.943	0.794	0.485	0.949	0.717
Reply Text + Comment Text	Stacked Bi-LSTM (GloVe)	0.597	0.908	0.939	0.657	0.556	0.941	0.638
Reply Text + Comment Text	Bi-LSTM (ELMo)	0.595	0.911	0.932	0.542	0.671	0.929	0.626
Question Text Only	Dense Stacked Bi-LSTM (ELMo)	0.595	0.931	0.925	0.505	0.763	0.916	0.712
Question Text Only	Dense Bi-LSTM (GloVe)	0.592	0.885	0.937	0.646	0.554	0.939	0.629
Reply Text + Comment Text	Dense Bi-LSTM (GloVe)	0.583	0.901	0.937	0.655	0.534	0.94	0.633
Reply Text + Comment Text	Dense Stacked Bi-LSTM (GloVe)	0.582	0.895	0.936	0.619	0.558	0.937	0.603
Reply Text	Bi-LSTM (ELMo)	0.582	0.92	0.924	0.492	0.722	0.916	0.654
Reply Text	Dense Bi-LSTM (GloVe)	0.581	0.894	0.936	0.623	0.554	0.937	0.623
Reply Text + Comment Text	Dense Stacked Bi-LSTM (ELMo)	0.58	0.902	0.932	0.569	0.601	0.931	0.601
Question Text Only	Dense LSTM (ELMo)	0.577	0.922	0.92	0.467	0.77	0.91	0.68
Reply Text	Dense LSTM (ELMo)	0.577	0.901	0.927	0.512	0.673	0.922	0.637
Reply Text	Dense Bi-LSTM (ELMo)	0.576	0.917	0.921	0.477	0.744	0.911	0.65
Question Text Only	Bi-LSTM (GloVe)	0.575	0.891	0.932	0.6	0.585	0.932	0.64
Reply Text + Comment Text	Stacked Bi-LSTM (ELMo)	0.574	0.906	0.931	0.563	0.592	0.93	0.594
Question Text Only	LSTM (GloVe)	0.573	0.871	0.936	0.65	0.521	0.939	0.612
Reply Text	Stacked Bi-LSTM (ELMo)	0.567	0.916	0.926	0.539	0.638	0.921	0.632
Reply Text + Comment Text	LSTM (ELMo)	0.566	0.888	0.928	0.539	0.62	0.925	0.555
Reply Text	Bi-LSTM (GloVe)	0.566	0.887	0.937	0.685	0.485	0.941	0.625
Reply Text	Stacked Bi-LSTM (GloVe)	0.563	0.89	0.932	0.582	0.55	0.932	0.61
Question Text Only	LSTM (ELMo)	0.561	0.925	0.915	0.444	0.768	0.903	0.684
Reply Text	Dense LSTM (GloVe)	0.557	0.872	0.932	0.613	0.528	0.933	0.59
Reply Text	LSTM (GloVe)	0.556	0.865	0.931	0.599	0.541	0.932	0.589
Reply Text + Comment Text	Dense LSTM (GloVe)	0.552	0.88	0.933	0.631	0.508	0.936	0.592
Question Text Only	Dense Stacked Bi-LSTM (GloVe)	0.552	0.87	0.925	0.568	0.589	0.921	0.615
Question Text Only	Dense LSTM (GloVe)	0.551	0.881	0.928	0.571	0.563	0.928	0.617
Reply Text + Comment Text	LSTM (GloVe)	0.545	0.88	0.933	0.639	0.488	0.937	0.593
Reply Text	LSTM (ELMo)	0.541	0.898	0.914	0.447	0.698	0.904	0.603
Reply Text	Dense Stacked Bi-LSTM (GloVe)	0.529	0.903	0.912	0.494	0.658	0.902	0.608

Table 6: Classification results for deep learning models on the high-agreement dataset: *confidence* $= 1.0$

Text Input	Feature	F1-score	AUROC	Weighted F1	Precision	Recall	Accuracy	AUPRC
Reply Text	word (1, 3)	**0.443**	0.782	0.823	0.549	0.373	0.836	0.481
Question Text Only	word (1, 3) + Sentiment + Writing-Style + Lexicon	0.432	0.764	0.81	0.471	0.4	0.816	0.443
Reply Text	word (1, 3) + Sentiment + Writing-Style + Lexicon	0.429	0.785	0.825	0.6	0.335	0.843	0.491
Question Text Only	char (3, 5) + Sentiment + Writing-Style + Lexicon	0.423	0.752	0.814	0.508	0.363	0.826	0.46
Question Text Only	word (1, 3)	0.422	0.764	0.815	0.519	0.357	0.828	0.46
Question Text Only	char (5, 5) + Sentiment + Writing-Style + Lexicon	0.419	0.753	0.813	0.503	0.359	0.825	0.457
Question Text Only	char (4, 4) + Sentiment + Writing-Style + Lexicon	0.417	0.762	0.816	0.531	0.345	0.83	0.464
Question Text Only	char (3, 5)	0.415	0.748	0.812	0.5	0.355	0.824	0.451
Reply Text	char (3, 5)	0.414	0.759	0.817	0.542	0.335	0.833	0.463
Reply Text	char (3, 5) + Sentiment + Writing-Style + Lexicon	0.412	0.764	0.815	0.531	0.336	0.831	0.467
Reply Text + Comment Text	word (1, 3) + Sentiment + Writing-Style + Lexicon	0.412	0.727	0.793	0.415	0.411	0.794	0.39
Reply Text	char (5, 5) + Sentiment + Writing-Style + Lexicon	0.409	0.767	0.819	0.57	0.319	0.838	0.471
Reply Text + Comment Text	word (1, 3)	0.409	0.786	0.818	0.561	0.323	0.836	0.466
Question Text Only	char (3, 3) + Sentiment + Writing-Style + Lexicon	0.408	0.759	0.814	0.526	0.336	0.829	0.456
Reply Text	char (4, 4) + Sentiment + Writing-Style + Lexicon	0.407	0.744	0.809	0.487	0.349	0.821	0.437
Question Text Only	char (4, 4)	0.403	0.739	0.803	0.456	0.361	0.812	0.432
Question Text Only	word (1, 1) + Sentiment + Writing-Style + Lexicon	0.4	0.765	0.813	0.525	0.323	0.83	0.46
Reply Text	char (4, 4)	0.4	0.742	0.807	0.482	0.342	0.82	0.434
Reply Text	char (5, 5)	0.399	0.727	0.801	0.446	0.362	0.809	0.402
Reply Text + Comment Text	char (4, 4) + Sentiment + Writing-Style + Lexicon	0.398	0.758	0.813	0.531	0.319	0.831	0.443
Question Text Only	char (5, 5)	0.394	0.744	0.808	0.495	0.328	0.823	0.448
Reply Text + Comment Text	word (1, 1) + Sentiment + Writing-Style + Lexicon	0.392	0.737	0.802	0.456	0.344	0.813	0.406
Reply Text + Comment Text	char (3, 5) + Sentiment + Writing-Style + Lexicon	0.391	0.729	0.798	0.438	0.354	0.807	0.399
Question Text Only	char (3, 3)	0.39	0.704	0.784	0.385	0.395	0.783	0.357
Reply Text	char (3, 3) + Sentiment + Writing-Style + Lexicon	0.389	0.71	0.782	0.38	0.399	0.78	0.369
Reply Text + Comment Text	Writing Style + Lexicon + Sentiment + GloVe	0.387	0.767	0.81	0.516	0.309	0.828	0.432
Reply Text + Comment Text	char (5, 5) + Sentiment + Writing-Style + Lexicon	0.387	0.734	0.8	0.446	0.342	0.81	0.403
Reply Text + Comment Text	char (3, 3) + Sentiment + Writing-Style + Lexicon	0.385	0.72	0.793	0.418	0.357	0.8	0.383
Question Text Only	Writing Style + Lexicon + Sentiment + GloVe	0.384	0.783	0.817	0.609	0.281	0.842	0.474
Reply Text + Comment Text	Writing Style + Lexicon + GloVe	0.384	0.768	0.809	0.515	0.307	0.828	0.434
Reply Text + Comment Text	word (1, 1)	0.384	0.73	0.8	0.449	0.336	0.811	0.398
Reply Text	word (1, 1) + Sentiment + Writing-Style + Lexicon	0.384	0.696	0.786	0.392	0.376	0.788	0.358
Question Text Only	word (2, 2) + Sentiment + Writing-Style + Lexicon	0.384	0.707	0.8	0.451	0.335	0.811	0.386
Reply Text	char (3, 3)	0.383	0.704	0.78	0.375	0.39	0.778	0.362
Question Text Only	Writing Style + Lexicon + GloVe	0.383	0.782	0.816	0.597	0.283	0.84	0.472
Reply Text + Comment Text	Fasttext	0.379	0.765	0.809	0.525	0.296	0.829	0.428
Reply Text	word (1, 1)	0.377	0.697	0.788	0.4	0.357	0.793	0.362
Reply Text	Writing Style + Lexicon + Sentiment + GloVe	0.377	0.785	0.813	0.571	0.282	0.837	0.465
Reply Text + Comment Text	char (5, 5)	0.377	0.718	0.796	0.436	0.332	0.807	0.389
Reply Text	Writing Style + Lexicon + GloVe	0.376	0.786	0.813	0.57	0.281	0.837	0.466
Question Text Only	word (1, 1)	0.375	0.697	0.787	0.395	0.357	0.791	0.352
Reply Text + Comment Text	char (3, 5)	0.374	0.72	0.792	0.415	0.342	0.8	0.386
Reply Text + Comment Text	char (3, 3)	0.371	0.714	0.789	0.405	0.344	0.796	0.376
Reply Text + Comment Text	char (4, 4)	0.367	0.751	0.805	0.5	0.291	0.824	0.433
Reply Text + Comment Text	RedditW2V	0.367	0.762	0.807	0.517	0.285	0.828	0.416
Question Text Only	Fasttext	0.365	0.775	0.811	0.58	0.266	0.837	0.459
Question Text Only	GloVe	0.362	0.78	0.812	0.594	0.261	0.839	0.464
Reply Text	word (2, 2) + Sentiment + Writing-Style + Lexicon	0.361	0.727	0.808	0.544	0.271	0.832	0.425
Reply Text + Comment Text	GloVe	0.356	0.762	0.804	0.511	0.273	0.827	0.424
Reply Text	Fasttext	0.355	0.781	0.809	0.569	0.258	0.835	0.455
Reply Text + Comment Text	word (2, 2) + Sentiment + Writing-Style + Lexicon	0.354	0.704	0.79	0.418	0.307	0.803	0.362
Question Text Only	word (3, 3) + Sentiment + Writing-Style + Lexicon	0.348	0.712	0.804	0.526	0.261	0.829	0.4
Reply Text	RedditW2V	0.345	0.778	0.806	0.556	0.25	0.833	0.438
Question Text Only	RedditW2V	0.342	0.776	0.806	0.559	0.247	0.834	0.436
Reply Text	GloVe	0.336	0.778	0.805	0.557	0.241	0.833	0.452
Reply Text	word (2, 2)	0.326	0.651	0.79	0.43	0.262	0.809	0.321
Reply Text + Comment Text	word (2, 2)	0.324	0.68	0.785	0.403	0.271	0.801	0.338
Question Text Only	word (2, 2)	0.324	0.676	0.791	0.438	0.257	0.812	0.36
Reply Text	word (3, 3) + Sentiment + Writing-Style + Lexicon	0.324	0.69	0.797	0.496	0.242	0.823	0.37
Reply Text + Comment Text	word (3, 3) + Sentiment + Writing-Style + Lexicon	0.294	0.752	0.798	0.575	0.198	0.833	0.431
Reply Text + Comment Text	GoogleW2V	0.291	0.735	0.787	0.435	0.219	0.813	0.362
Question Text Only	GoogleW2V	0.279	0.752	0.792	0.511	0.192	0.826	0.394
Question Text Only	word (3, 3)	0.278	0.62	0.79	0.491	0.194	0.823	0.314
Reply Text	GoogleW2V	0.26	0.754	0.786	0.481	0.178	0.822	0.384
Reply Text	word (3, 3)	0.25	0.592	0.783	0.46	0.172	0.819	0.274

Table 7: Classification results for Logistic Regression on the complete dataset: *confidence* $\in \{0.6, 0.8, 1.0\}$

Text Input	Feature	F1-score	AUROC	Weighted F1	Precision	Recall	Accuracy	AUPRC
Question Text Only	Writing Style + Lexicon + Sentiment + GloVe	**0.618**	0.906	0.942	0.671	0.574	0.944	0.648
Question Text Only	Writing Style + Lexicon + GloVe	0.616	0.907	0.942	0.672	0.572	0.944	0.65
Question Text Only	char (3, 5) + Sentiment + Writing-Style + Lexicon	0.614	0.916	0.944	0.753	0.521	0.949	0.672
Question Text Only	char (5, 5) + Sentiment + Writing-Style + Lexicon	0.612	0.917	0.944	0.747	0.521	0.948	0.666
Question Text Only	char (4, 4) + Sentiment + Writing-Style + Lexicon	0.606	0.917	0.943	0.733	0.518	0.947	0.668
Question Text Only	word (1, 3) + Sentiment + Writing-Style + Lexicon	0.6	0.917	0.941	0.711	0.521	0.945	0.665
Question Text Only	GloVe	0.594	0.897	0.938	0.639	0.558	0.94	0.611
Question Text Only	Fasttext	0.59	0.907	0.938	0.648	0.543	0.941	0.623
Question Text Only	word (1, 1) + Sentiment + Writing-Style + Lexicon	0.582	0.907	0.936	0.627	0.545	0.938	0.643
Question Text Only	char (3, 3) + Sentiment + Writing-Style + Lexicon	0.581	0.905	0.938	0.663	0.521	0.941	0.629
Reply Text	word (1, 3) + Sentiment + Writing-Style + Lexicon	0.567	0.93	0.939	0.745	0.461	0.945	0.654
Question Text Only	word (2, 2) + Sentiment + Writing-Style + Lexicon	0.566	0.896	0.937	0.671	0.49	0.941	0.605
Reply Text	word (1, 1) + Sentiment + Writing-Style + Lexicon	0.558	0.913	0.936	0.677	0.481	0.941	0.62
Question Text Only	word (1, 3)	0.554	0.918	0.935	0.651	0.483	0.939	0.579
Question Text Only	char (4, 4)	0.552	0.891	0.934	0.649	0.483	0.939	0.581
Reply Text	char (3, 5) + Sentiment + Writing-Style + Lexicon	0.549	0.9	0.934	0.646	0.479	0.938	0.598
Reply Text	char (4, 4) + Sentiment + Writing-Style + Lexicon	0.549	0.925	0.937	0.765	0.432	0.945	0.654
Reply Text	Fasttext	0.545	0.898	0.932	0.618	0.49	0.936	0.58
Reply Text	word (1, 3)	0.544	0.925	0.935	0.688	0.452	0.941	0.608
Reply Text	Writing Style + Lexicon + GloVe	0.539	0.897	0.931	0.589	0.499	0.933	0.586
Reply Text	char (5, 5) + Sentiment + Writing-Style + Lexicon	0.539	0.926	0.936	0.771	0.419	0.944	0.651
Reply Text	Writing Style + Lexicon + Sentiment + GloVe	0.538	0.897	0.931	0.597	0.492	0.934	0.587
Reply Text + Comment Text	word (1, 3)	0.537	0.888	0.926	0.526	0.552	0.925	0.5
Reply Text + Comment Text	word (1, 1) + Sentiment + Writing-Style + Lexicon	0.533	0.915	0.934	0.702	0.433	0.941	0.597
Reply Text	char (3, 5)	0.53	0.889	0.932	0.643	0.452	0.937	0.57
Question Text Only	char (3, 5)	0.53	0.917	0.936	0.783	0.404	0.944	0.646
Reply Text	char (3, 3) + Sentiment + Writing-Style + Lexicon	0.529	0.909	0.932	0.646	0.45	0.937	0.606
Reply Text	word (2, 2) + Sentiment + Writing-Style + Lexicon	0.529	0.909	0.934	0.735	0.415	0.942	0.612
Question Text Only	word (1, 1)	0.527	0.888	0.93	0.602	0.47	0.933	0.56
Reply Text	GloVe	0.526	0.889	0.93	0.613	0.463	0.935	0.567
Reply Text + Comment Text	word (1, 3) + Sentiment + Writing-Style + Lexicon	0.526	0.883	0.924	0.508	0.545	0.922	0.501
Reply Text + Comment Text	Fasttext	0.525	0.876	0.924	0.512	0.541	0.923	0.516
Reply Text + Comment Text	char (5, 5) + Sentiment + Writing-Style + Lexicon	0.523	0.89	0.931	0.648	0.439	0.937	0.569
Reply Text + Comment Text	char (3, 5) + Sentiment + Writing-Style + Lexicon	0.521	0.925	0.934	0.756	0.399	0.942	0.637
Question Text Only	char (3, 3)	0.519	0.889	0.93	0.62	0.448	0.935	0.575
Reply Text + Comment Text	Writing Style + Lexicon + GloVe	0.519	0.879	0.924	0.515	0.525	0.923	0.528
Reply Text	word (1, 1)	0.517	0.898	0.931	0.658	0.43	0.938	0.578
Reply Text + Comment Text	char (3, 3) + Sentiment + Writing-Style + Lexicon	0.517	0.911	0.932	0.676	0.419	0.938	0.603
Reply Text + Comment Text	char (4, 4) + Sentiment + Writing-Style + Lexicon	0.515	0.888	0.93	0.636	0.435	0.936	0.571
Reply Text	char (3, 3)	0.512	0.881	0.926	0.565	0.468	0.93	0.549
Reply Text + Comment Text	Writing Style + Lexicon + Sentiment + GloVe	0.511	0.878	0.922	0.503	0.521	0.922	0.522
Reply Text + Comment Text	word (2, 2) + Sentiment + Writing-Style + Lexicon	0.507	0.911	0.932	0.739	0.386	0.941	0.602
Reply Text	char (5, 5)	0.506	0.881	0.928	0.61	0.435	0.934	0.549
Question Text Only	RedditW2V	0.505	0.889	0.925	0.56	0.461	0.928	0.506
Question Text Only	char (5, 5)	0.505	0.909	0.932	0.764	0.379	0.942	0.621
Reply Text	char (4, 4)	0.504	0.879	0.928	0.599	0.437	0.933	0.55
Reply Text + Comment Text	GloVe	0.504	0.872	0.922	0.505	0.507	0.922	0.516
Question Text Only	word (3, 3) + Sentiment + Writing-Style + Lexicon	0.499	0.891	0.93	0.688	0.395	0.937	0.591
Reply Text + Comment Text	char (5, 5)	0.493	0.874	0.927	0.627	0.406	0.934	0.533
Reply Text + Comment Text	char (3, 5)	0.489	0.876	0.926	0.612	0.408	0.933	0.529
Reply Text	RedditW2V	0.488	0.877	0.924	0.556	0.441	0.928	0.483
Reply Text + Comment Text	word (1, 1)	0.487	0.885	0.927	0.616	0.404	0.933	0.518
Reply Text + Comment Text	RedditW2V	0.483	0.875	0.92	0.503	0.466	0.921	0.479
Reply Text + Comment Text	char (4, 4)	0.482	0.882	0.926	0.61	0.399	0.933	0.552
Reply Text	word (3, 3) + Sentiment + Writing-Style + Lexicon	0.472	0.847	0.925	0.614	0.388	0.932	0.505
Reply Text + Comment Text	char (3, 3)	0.47	0.901	0.927	0.715	0.351	0.937	0.572
Question Text Only	Writing Style + Lexicon + Sentiment	0.457	0.88	0.926	0.697	0.342	0.936	0.543
Question Text Only	GoogleW2V	0.456	0.874	0.92	0.531	0.399	0.925	0.453
Question Text Only	Writing Style + Lexicon	0.446	0.87	0.925	0.72	0.324	0.937	0.518
Question Text Only	word (2, 2)	0.443	0.832	0.923	0.62	0.346	0.932	0.446
Reply Text	word (2, 2)	0.443	0.802	0.921	0.58	0.36	0.929	0.419
Reply Text + Comment Text	word (3, 3) + Sentiment + Writing-Style + Lexicon	0.439	0.864	0.92	0.569	0.36	0.927	0.476
Reply Text + Comment Text	Writing Style + Lexicon + Sentiment	0.409	0.877	0.92	0.658	0.298	0.932	0.49
Reply Text + Comment Text	word (2, 2)	0.397	0.81	0.915	0.534	0.318	0.924	0.398
Reply Text + Comment Text	GoogleW2V	0.393	0.844	0.903	0.385	0.406	0.901	0.371

Table 8: Classification results for Logistic Regression on the high-agreement dataset: *confidence* $= 1.0$