Explain your predictions: A Step Towards Reasoning in Natural Language Processing

Murali Manohar Kondragunta **Duygu Bayram**
{5397294, 5416752}

Abstract

Recent work showed that Natural Language Explanation (NLE) is achievable with enough computation and resources. In this work, we explore the possibility of including reasoning in an offensiveness detection system and see whether it improves or degrades the system. To build a model that explains its predictions, we create an artificial explanation dataset with the list of offensive words in the sentence using online lexicons. From the experiments on the T5 language model, we observed that adding explanations did not improve the model's ability in detecting offensiveness. However, we show that we can successfully incorporate explanations into the model without disturbing the model's performance on downstream tasks, in our case, offensiveness detection. Moreover, we demonstrate that challenging linguistic structures do not degrade the performance of the model. The code is publicly released on Github[1].

1 Introduction

With the rise of social media sites, hate mongering has been rampant all over the internet (Schmidt and Wiegand, 2017), bringing about a growing number of shared tasks (Waseem and Hovy, 2016a; Fersini et al., 2018a; Fersini et al., 2018b; Fersini et al., 2020; Guest et al., 2021; Basile et al., 2019). OffensEval (Zampieri et al., 2019a) is one such shared task where the goal is to identify whether a given tweet is offensive or not, and whether it is targeted towards a group or an individual. To this end, they released the OLID (Offensive Language Identification Dataset) dataset that we use in our models.

Deploying a hate speech detection model into the real world without any explanation or reasoning over the prediction does not contribute to the user experience (Epstein et al., 2021). Therefore, we intend to build an explainable model that gives reasons for its predictions, and to evaluate whether forcing the model to generate explanations improves its performance in offensiveness detection.

1.1 Prompting the models

Despite the success of neural models in natural language processing tasks, the huge set of parameters make their interpretation difficult. So, neural models remained a black-box for long time making them hard to reason over the predictions made. However, recent studies (Ling et al., 2017; Cobbe et al., 2021; Sap et al., 2019; ElSherief et al., 2021; Huang et al., 2022) showed that, with enough resources, Natural Language Explanation (NLE) can be achieved. For instance, Wei et al. (2022) showed passing a chain of intermediate steps with the answer can help the model improve its answering capability along with the additional benefit of reasoning the prediction. Despite achieving comparably good results with just 8 examples, it is not feasible to use a large language model like PALM (Chowdhery et al., 2022) (540 billion parameters) in production. Huang et al. (2022) follow the same idea of chain of thought prompting for NLE in hate speech detection, but with smaller generative models like GPT-Neo (Black et al., 2021), OPT (Zhang et al., 2022), T5 (Raffel et al., 2019), and BART (Lewis et al., 2020), and compensate for their fewer parameters with a large dataset of 6,500 instances with explanations.

[1] https://github.com/gitlost-murali/explain-preds

1.2 Our work

In contrast to both of these systems, we neither have access to the large language models nor to the manual explanations of the OLID dataset (Zampieri et al., 2019a). Therefore, we create an artificial explanation dataset for OLID to train an explainable system.

In this work, we explore the feasibility of explanation in text classification problems and its role in improving the model's answering capability using an artificial dataset. Specifically, we train a seq2seq model to predict the label along with the explanation. Achieving this would require a manually annotated dataset where the explanations are laid out for the model to understand human reasoning. In our case, we approximate this process by creating an artificial dataset where we pass the filtered offensive words in the sentence as an explanation for the label. Although this is an approximation of the task, we have seen comparable results and we consider it as initial steps towards reasoning in Natural language understanding. Further, we use our prompts to test how our model fares with linguistically challenging sentences.

We perform the experiments in three phases:

1. In the first phase, we evaluate different baselines (discriminatory models) to estimate the difficulty of the task.

2. In the second phase, we evaluate if we can successfully pivot the classification problem into a seq2seq one. In this step, we limit the target sequence to label definitions (verbalizers) i.e. offensive and not offensive.

3. In the third phase, we create the artificial explanation dataset (shown in Figure 2). Later, we train the seq2seq model on the artificial dataset where the model must predict both the label and the explanation behind that label. Here, we observe if we can incorporate explanations without performance loss. As the choice of prompts can greatly influence the performance of a prompting based model (Zhao et al., 2021), we experiment with linguistically challenging prompts which makes it harder for the model to learn new patterns. This way, we ensure that the potential positive results are not due to chance or luck while designing prompts, and that the model performs well despite difficult prompts.

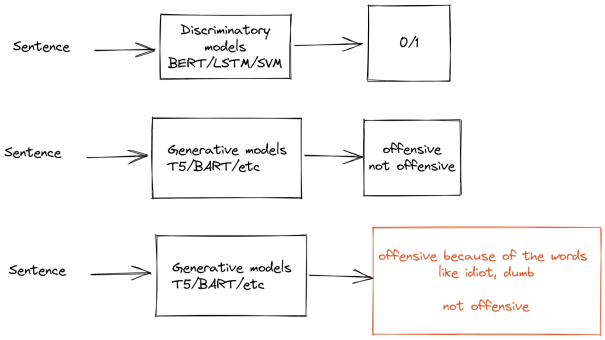

Figure 1: Three phases of approaching the problem. The order of phases can be construed in a top-down manner. For instance, discriminatory models being the first phase.

2 Related Work

2.1 Offensive speech

Offensive speech is near-unavoidable in online spaces due to the anonymous and fast-paced nature of interactions. These interactions can be grouped in several different ways, such as aggression, offensiveness, hate speech, cyberbullying, and toxicity (Zampieri et al., 2019b). It is important to find ways to moderate targeted negative speech for user safety and satisfaction, and Natural Language Processing (NLP) tools have been shown to be helpful in this regard (Waseem and Hovy (2016b), Founta et al. (2018), Davidson et al. (2017)).

There has been several events aiming to improve the available tools around classifying negative speech such as OffensEval-2019 (Zampieri et al., 2019b), TRAC (Kumar et al., 2018), Kaggle Toxic Comment Classification Challenge[2], and HASOC-2019 (Mandl et al., 2019). OffensEval-2019 invited many different approaches from participants, most have used pre-trained language models, with BERT (Devlin et al., 2018) consistently producing the top-performing results (Zampieri et al., 2019b). Others have built ensembles of various other deep learning models such as CNN (LeCun et al., 1998), BiLSTM (Zhou et al., 2016), and BiGRU (Cho et al., 2014). OffensEval-2020 showed similar leaning towards language models, with most participants choosing BERT, RoBERTa (Liu et al., 2019b), mBERT (Devlin et al., 2018), and XLM-RoBERTa (Conneau et al., 2019), and others employing CNNs, RNNs

[2]https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge

(Rumelhart et al., 1985), and GRUs (Zampieri et al., 2020). Due to their demonstrated high performance in these tasks, we choose LSTM, BERT and RoBERTa as our baseline models.

In both events, participants have used fastText (Bojanowski et al., 2017), GloVe (Pennington et al., 2014), or word2vec (Mikolov et al., 2013) word embeddings, with some participants using offensive word lexicon approaches. The use of lexicons in one of the top-performing models of OffensEval (Seganti et al., 2019) is promising, as it shows that offensive words can be helpful features for classification on the OffensEval dataset, which we are also using for our models. As we are using the lexicon for explanation in later stages, this support is important for our work.

2.2 Natural language explanation

The field of natural language explanations has been pioneered by Ling et al. (2017) when they suggested using natural language rationales as intermediate steps to solve math problems. Cobbe et al. (2021) extended Ling et al. (2017)'s work by curating a larger dataset and fine-tuning a pretrained language model instead of training it from scratch. However, curating such explanations is costly. To address the drawbacks of previous works, Wei et al. (2022) leveraged the few-shot prompting method from Brown et al. (2020) and a new prompting approach called *chain-of-thought prompting*, where a series of intermediate natural language explanations are passed to guide the model towards the final answer.

When deploying hate speech detection models, Epstein et al. (2021) showed that providing explanations for the predictions improved user experience and system efficacy. To stop harmful posts at the source, ElSherief et al. (2021), Sap et al. (2019) and Mathew et al. (2020) provided datasets with the implied meanings of the text along with the details of target group. While ElSherief et al. (2021) and Sap et al. (2019) relied on the GPT-2 model (Radford et al., 2019) without any prompting, Huang et al. (2022) followed the chain of thoughts prompting methodology and benchmarked their framework on autoregressive models like GPT-Neo, OPT and Seq2Seq models like BART and T5. However, all the models mentioned above leveraged a dataset with manual annotations. In our case, the dataset does not offer explanations for the input text. So, we create an artificial dataset with synthetic explanations using a template based approach that mentions the offensive words in the text as an explanation.

3 Data

The Offensive Language Identification Dataset (OLID) is collected from Twitter API by searching keywords that are frequently used in offensive tweets, such as 'she is', 'he is', and 'to:BreitBartNews'. OLID contains 14,100 annotated English tweets divided into a training partition of 13,240 tweets and a test set of 860 tweets. In our experiments, the training data was further split into a development set of 1,000 tweets and a training set of 12,240 tweets. Each tweet was annotated on the basis of a hierarchical three-level model:

A - Whether the tweet is offensive (OFF) or not (NOT).

B - Whether the tweet is targeted (TIN) or untargeted (UNT).

C - If the target is an individual (IND), a group (GRP) or other (OTH; e.g., an issue or an organisation).

As we focus on subtask A, our use of the dataset was preprocessed to contain only two labels, 'OFF' and 'NOT'. Table 1 shows the distribution of each class. OLID does not have an equal number of offensive and non-offensive tweets. Only about 30% of the tweets are marked offensive, making the task more challenging with an imbalanced dataset. Another feature to note is that the Twitter IDs in the tweets were replaced by the placeholder '@USER' and the URLs were replaced by 'URL' to ensure normalization and anonymity.

	Train	**Dev**	**Test**	**Total**
OFF	4,048	352	240	4,640
NOT	8,192	648	620	9,460
All	12,240	1,000	860	14,100

Table 1: Distribution of the tweets in the dataset per label. The first column denotes the two categories tweets belong to, offensive or not. The next three columns represent the three sets we split the original data set into, the training set, the development set, and the test set.

4 Methodology

4.1 Preprocessing

This section briefly describes how we preprocessed the content of tweets. Following Liu et al. (2019a), who ranked first in the sub-task A of SemEval 2019, we substitute the emojis with their corresponding English phrases[3] and use a word segmentation tool[4] to split the hashtags back to words. Moreover, emojis have been shown to be useful in sentiment analysis tasks (Ayvaz and Shiha, 2017) and we believe converting them to their semantic meanings would be informative. The same belief goes with the hashtags. A hashtag may consist of multiple words, including offensive words, so we applied word segmentation on hashtags. Additionally, we replaced the placeholder 'URL' with 'http', as the term 'URL' lacks an embedding representation in some pre-trained embeddings and models (Liu et al., 2019a). We also set a limit on the number of consecutive '@USER' tags to three times to avoid redundancy, following the example of (Liu et al., 2019a). As a final step, we converted all the texts into lowercase.

4.2 Discriminatory Baselines

Baseline models considered in the phase-1 (Figure 1) are SVM (Boser et al., 1992), LSTM (Hochreiter and Schmidhuber, 1997), BERT, and RoBERTa. For neural models like LSTM, BERT, and RoBERTa, we stick to the preprocessing mentioned in Section 4.1.

SVM: We performed grid search to find the best hyperparameters for Linear SVM[5]. We mainly looked for the best C value for margins, and whether using inverse weights for the class frequencies was helpful to account for the imbalanced dataset. We manually tried different vectorizers with different preprocessing methods, the results of our experiments can be viewed in Appendix, Table A, also showing the best performing vectorizer. The best performing hyperparameters for our baselines can be viewed in Appendix, Table 6.

For SVM, preprocessing steps such as the removal of stopwords and replacing numbers with 'NUM' tokens yielded better results; however, we decided not to use these steps for neural language models as this information can aid them. Our other experiments showed keeping the URL tags produced better results, along with converting emojis to natural language, so we kept these steps in our other models. Removing hashtags did not have an effect, we reasoned this is because they were unsegmented in our SVM runs, so we implemented hashtag segmentation in the following models, inspired by (Liu et al., 2019a).

LSTM: For our LSTM baseline, we use a two layer Bidirectional LSTM followed by a dense layer with `ReLU` activation function and a dropout of 0.2 between them. We leveraged fastText embeddings (Grave et al., 2018) because of their coverage. We train the model for 20 epochs with a learning rate of 1e-4, a batch size of 32 and max sequence length of 100 tokens. To avoid overfitting, we added an `EarlyStopping` mechanism with a patience of 3 epochs to the training. The baseline is implemented using Keras.

BERT: We implemented the BERT baseline using the Huggingface `bert-base-uncased`[6] model with the preprocessing steps described in Section 4.1. The model is trained for 5 epochs with a learning rate of 5e-5, batch size of 16, and the Adam optimizer. We also implemented the same mechanism of `EarlyStopping` with a patience of 3.

RoBERTa: For RoBERTa[7], we implemented the same steps as BERT and experimented with different values of learning rate, batch size, and maximum sequence length. The best performing model had a learning rate of 5e-5, a batch size of 32, and a maximum sequence length of 150.

Neural models like LSTM, BERT, and RoBERTa are implemented using Tensorflow (Abadi et al., 2015) and Keras (Chollet and others, 2015) frameworks.

4.3 Our approach

Typically, text classification tasks are performed in a discriminatory way: the model receives text and the corresponding labels, and the model subsequently learns to predict the correct labels for unseen text data. However, the limitation of these models is that they only give us a label as the out-

[3] https://github.com/carpedm20/emoji

[4] https://github.com/grantjenks/python-wordsegment

[5] https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html

[6] https://huggingface.co/bert-base-uncased?text=The+goal+of+life+is+%5BMASK%5D.

[7] https://huggingface.co/roberta-base?text=The+goal+of+life+is+%3Cmask%3E.

put, and are therefore insufficient if we want to explain why the model chose a specific label.

Instead of following the typical approach, we decided to recast it as a machine translation task which allows us to have a text-to-text model that will help us output an explanation in the later stages. Our model receives the input text, and sequentially outputs a natural language sentence. In other words, the model learns the correct translation (which expresses the label and the reason) of a given sentence (which is the text input we want to classify).

Since T5 is pre-trained on tasks that are pivoted to a seq2seq framework, we used it as the base generative model for our experiments. For fair comparison with RoBERTa, we take T5-base[8] (220 Million parameters) although RoBERTa is still smaller with only 123 Million parameters.

For both of our phases, we trained the model with a learning rate of 1e-4 for 5 epochs with an `EarlyStopping` mechanism set to a patience of 3 epochs. Both phases are using `PytorchLightning` framework (Falcon et al., 2019).

Using this model, we experimented with different variations of the output (prompts) to analyze how the model handles linguistic challenges, as detailed in Section 4.3.2. For the output type that contained an explanation, we created a lexicon for offensive words, as detailed in Section 4.3.1.

4.3.1 Dataset creation

To approximate human level explanations, we decided to create a lexicon of offensive words. This allows us to incorporate human reasoning to an extent, as we are borrowing a partial knowledge base of what makes an expression offensive. By doing this, we are able to provide an explanation for our labels by pointing out the offensive words in a text.

Figure 2 illustrates the template generation process. Each template has two parts. Label definition (verbalizer) and the explanation. Label verbalizer refers to converting the label (”OFF”/”NOT”) to a natural language definition (”offensive” for ”OFF” in Figure 2). The explanation template has a template sentence such as ”because of the words like”, followed by the list of offensive words in the sentence.

To build our offensive lexicon, we merged 6 existing lists of offensive words, Harrassment Corpus and Offensive/Profane Word List (Rezvan et al., 2018), Hurtlex (Bassignana et al., 2018), List of Dirty Naughty Obscene and Otherwise Bad Words[9], Profone Words[10], and Google Profanity Words[11].

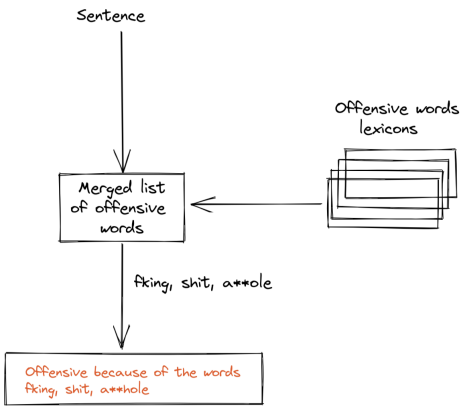

Figure 2: Template creation process

We dropped the *inclusive* category from Hurtlex (Bassignana et al., 2018), as it includes all potentially relevant words for offensiveness, and as such not removing it initially caused the model to give us non-offensive words. Because our task does not require categories or attributes apart from the words, we removed the categorization system existing in some lexicons. During merging, we also lowercased all words and removed duplicates (4,142 in total), resulting in a lexicon of 5,596 words. The distribution of words in each list can be seen in Table 4.3.1. We then find the occurrence of these words in each input text in our dataset, and merge the corresponding occurrences to our existing dataset as an explanation.

4.3.2 Different prompts

As the choice of templates and label verbalizers can greatly influence the performance of a prompt-learning pipeline (Zhao et al., 2021), we experiment with different prompts.

Our first prompt level does not contain an explanation, here we experimented with outputs that gave us some representation of the label, either as binary numbers (0/1), as label text (“offensive”/”non-offensive”), or as a natural language sentence (“This comment is offensive/not

[8] `https://huggingface.co/t5-base?text=My+name+is+Sarah+and+I+live+in+London`

[9] https://github.com/LDNOOBW/List-of-Dirty-Naughty-Obscene-and-Otherwise-Bad-Words

[10] https://github.com/zacanger/profane-words

[11] https://github.com/coffee-and-fun/google-profanity-words

Lists	Words
Harrassment-Corpus	723
Hurtlex *(without inclusive)*	3,360
Offensive/Profane Word List	1,382
LDBNOOBW	402
Profane Words	2,914
Google Profanity Words	957
Duplicates	4,142
Final Lexicon	5,596

Table 2: Count of items available in each existing lexicon, along with number of duplicates that were removed during merging, and the count of items available in our final lexicon.

offensive."). This allows us to test the effect of using natural language on the model performance.

In the second level, we incorporate our explanations to the output, and try different sentences of varying lengths, representing typical linguistic challenges. To observe how the model performs under challenging natural language environments, we have curated prompts for syntactic and semantic difficulties, these prompts are listed in Table 4.3.2.

Semantics: We chose a sentence containing an anaphor (Prompt-2) to investigate how the model may handle semantic pressures. Anaphoras are also known challenges in NLP applications (Sukthanker et al., 2020), so we found it worth to include. Although, since the nature of our task does not involve multiple subjects where the model may provide the wrong pronoun, we consider the possibility that our model will not have a difficulty with this prompt.

Syntax: To represent syntactic difficulty, we chose a Garden-Path sentence (Prompt-1) to start with. As Garden-Path sentences contain word occurences that may be high in surprisal (in other words, low in probability), and are difficult to parse for humans (Hale, 2001), it is possible that our model might struggle here. Additionally, we included a passive voice (Prompt-4) as these sentence structures contain NP-movements and are therefore more complicated than the typical active voice correspondence of the same sentence, as can be seen in Figure 3. To account for another syntactic difficulty, we chose a question (Prompt-3), which contains both Wh-movement4, and Long Distance Dependency (LDD), a structure that is often not preferred by humans in natural language as it strains memory (Liu et al., 2017).

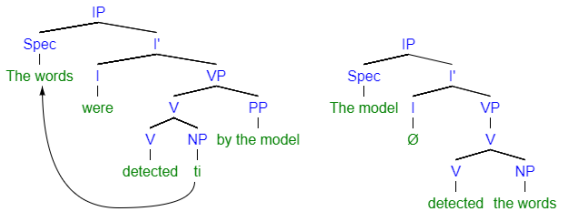

Figure 3: Syntactic structures of passive (left) and active (right) voice versions of the same sentence.

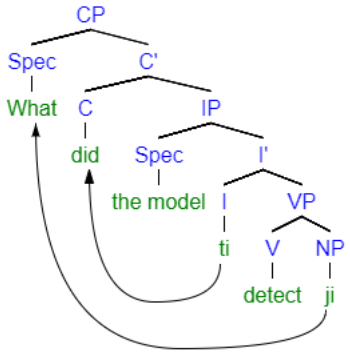

Figure 4: Syntactic structure of a question showing Wh-movement and Long Distance Dependency.

5 Results & Discussion

5.1 Baselines

SVM: We implemented a Linear SVM model as an initial baseline, and optimized the hyperparameters and vectorizers. We also experimented with preprocessing to gain insight on how to proceed for other models. The process of these experiments are discussed further in Section 4.2. Our main takeaways were that keeping 'URL' tags, converting emojis to natural language phrases, and segmenting hashtags were helpful. The best performing model used an n-gram vectorizer with a range of (1, 3) with Bag of Words, and the hyperparameters were set as C = 0.01 with balanced class weights to account for imbalanced data. A Macro-F1 score of 74.9 on the test set shows that this is a strong baseline.

LSTM: When trained with the best hyperparameters (Table 5.1) obtained from hyperparameter search, we found the model to perform better than the SVM baseline with a macro F1 score of

nr	category	offensive	not offensive	explanation
1	Garden-Path	"the model observed classified offensive "	"the model observed classified not offensive "	"since the following words showed up "
2	Anaphora	"we had several words that rendered this offensive "	"we had no words that rendered this offensive"	"they were "
3	Question	"which words made us decide this is offensive, you ask? "	"which words made us decide this is not offensive you ask "	"here you go "
4	Passive Voice	"the provided sentence may be interpreted as offensive by some users"	"the provided sentence may not be found offensive by most users"	"as certain offensive words occur such as"

Table 3: The linguistic challenge prompts provided to test model performance. Number indicates Prompt-No as referred to in other sections. The category points to the type of challenge. A sentence fragment is provided for the labels and the explanation.

Model	Macro-F1
SVM	74.9
LSTM	77.23 +_ 1.02
BERT	**80.3 +_ 0.71**
RoBERTa	75.56 +_ 6.8

Table 4: Macro-F1 results of baselines on the test set. The standard deviation is reported from 3 runs of each experiment.

77.23 (Table 5). This can be attributed to the usage of semantic features like embeddings and sequential processing of the sentences. It can also be observed that the Macro-F1 score difference between SVM and LSTM baselines is not much (74.9 vs 77.23). We conjecture that this is due to the strong presence of offensive words in offensive texts.

BERT & RoBERTa: For BERT & RoBERTa, we experimented with learning rate, batch size, and maximum sequence length to optimize the model. The best performing model for both versions had a learning rate of 5e-5, a batch size of 32, but a maximum sequence length of 150 for RoBERTa and 200 for BERT. When trained for 5 epochs, RoBERTa got 75.56% and BERT got 80% Macro F1-score on the test set. We expect the generative models to match the scores of BERT to make sure that the model performance does not deteriorate when explanations are added to the output.

5.2 Prompting results

We run two phases of experiments for T5, with explanation, and without explanation. In the first phase, we see if the task can be recast without loss in performance, and observe the effects of verbalizers. In the second phase, we add our explanations to the model, and prepare our prompts from linguistic difficulties to analyze the model's linguistic performance, and see if the overall performance holds up.

5.2.1 Recasting and effects of verbalizers

As a first step, we check if the task can be successfully pivoted to seq2seq framework. Therefore, we only pass the label definition or verbalizer to the model. As seen in Table 5, we found there is no loss of information from the conversion.

In this initial phase, we also experimented with different label verbalizers as they can impact the system's performance greatly (Zhao et al., 2021). Despite looking more natural than other label verbalizers, we found verbalizer (*the comment is offensive/not-offensive*) to not perform any different than other verbalizers. We conjecture that this is due to the high vocabulary overlap between the labels' verbalizers.

5.2.2 Explanations and linguistic performance

In the second phase, we add explanations to the template and evaluate their impact on the model's overall performance, along with analyzing the model's linguistic performance on our prompts.

Effect of adding explanations: We extract the label verbalizer from the generated explanation and compare the predicted label verbalizer with that of the ground truth one to calculate Macro-F1 score. To check how close the explanations are to that of the ground truth, we compare them using the BLEU metric (Post, 2018)[12]. From the experiments (Table 5), we observed that we can retain the model's performance despite adding the task overhead by adding explanations.

Linguistic performance: First, we ask if the presence of linguistic complexity impairs the

[12] https://github.com/mjpost/sacrebleu

Model	Prompts	Macro-F1	BLEU
T5	offensive/not offensive	**79.97 +_ 0.7**	
	the comment is offensive/not offensive	79.70 +_ 0.78	
	0/1	79.52 +_ 0.95	
T5 with explanations	prompt-1	78.93 +_ 0.9	81.06 +_ 1.9
	prompt-2	**79.64 +_ 0.45**	87.8 +_ 0.64
	prompt-3	78.53 +_ 0.465	**89.38 +_ 0.052**
	prompt-4	79.51 +_ 0.6	80.19 +_ 0.36

Table 5: Summary of T5 Macro-F1 and BLEU scores on the test set. Prompt numbers refer to the prompts described in Table 3. The standard deviation is reported from 3 runs of each experiment.

model performance. From Table 5, we see that the results are on par with those of other prompts and our baselines.

To analyze the linguistic performance of the model, we look at the slight drop in performance in Prompt-1 (Garden-Path), and Prompt-3 (Question). A potential explanation is the high surprisal effect of Garden-Path sentences, along with the Long Distance Dependency observable in the provided English question (See Section 4.3.2). However, the scores are not different enough to make this claim with confidence.

On the other hand, Arehalli et al. (2022) found lexical predictability is insufficient to explain the Garden-Path effect, and implementing syntactic predictability in models produces higher effects. This could explain our results, showing that our model is not sensitive to syntax. Prompt-3 (Question) could further support this idea, as the lack of effect can be explained by the commonality of such questions, despite the uncommonality and complexity of LDDs in syntactic structures.

An unexpected result is the large gap between the Macro-F1 and BLEU scores, as well as higher BLEU scores in Prompt 2 and 3 compared to the others (Table 5). We initially interpreted this to be due to length; however, Prompt-4 is longer than Prompt-2, and does not present similar results. Therefore, we conjecture this is due to the large amount of shared words between the 'offensive' and 'not offensive' labels in these prompts. This can be taken into consideration in future work regarding prompt engineering.

Overall, the model shows to be successful even with potential linguistic challenges. The use of prompt engineering in classification tasks for including explanations seems promising.

6 Error Analysis

We look closer at the errors made in Prompt-3 as it has the lowest Macro-F1 score in the T5 model. From the 149 errors detected, approximately 57% (85) are wrongly classified as offensive, while the remaining are wrongfully classified as not offensive.

Wrong 'not offensive' predictions: In the case of 'not offensive' misclassifications, the model does not provide an offensive words list due to the label. However, by looking at ground truth explanations, we can see which offensive words were extracted from the sentences.

Here, we observe that the explanation contains offensive words for most of these inputs, although a considerable portion of 37.5% (24) of the explanations are blank (Appendix, Table 7, 1a). This large lack of available explanations for the mislabeled inputs likely accounts for the error. Out of the remaining 40 mispredictions with offensive words, 6 (15%) of the explanations contain duplicate words (Appendix, Table 7, 1b). As we found very few occurences of this in the case of 'offensive' misclassifications, we think this factor could be contributing to the error. Furthermore, we observe that many of the offensive words provided are not necessarily offensive (Appendix, Table 7, 1c), which could be a contributing factor. This would point to limitations in the lexicon causing some of these errors.

Wrong 'offensive' predictions: Most of the errors are 'offensive' misclassifications, hinting that the model leans towards this label. In this case, the ground truth explanation is blank, as the true label is 'not offensive'. However, when we look at model predictions, we see that nearly all of them

contain an offensive word as an explanation, with only 5 instances being blank (Appendix, Table 7, 2a). As we are using explanations during training, the detection of offensive words in these inputs explains the error.

Considering this, we look at whether the retrieved words would be judged offensive by a human, to see if the issue is caused by the lexicon. While some still have the issue of being non-offensive (Appendix, Table 7, 2b), most are either offensive (Appendix, Table 7, 2c) or difficult to judge without context (Appendix, Table 7, 2d). As such, we see the limitations of using lexicons for offensiveness classification here, as the ability to understand context may change the label judgement for human annotations.

7 Conclusion

Offensive speech is a widespread issue on social media that affects user experience. To combat this issue, many NLP tools have been developed to detect similar targeted negative speech. However, these models are largely black-boxes, and we do not see the reasoning behind their decisions. To approximate reasoning, we borrow from human understanding and build an offensive word lexicon and recast the task as a machine translation task to provide an explanation for our classifications. We test our model with various prompts to see if the addition of natural language impairs classification, and to observe how the model performs under certain prompts with linguistic difficulties.

Our results show that the task can be recast as a seq2seq approach and be augmented with explanations without loss in the performance. Although previous findings suggest that label prompts can change the results greatly (Zhao et al., 2021), our results do not reflect this. However, the model continues to perform well on our prompts overall, demonstrating that the model is able to handle linguistic challenges, despite potentially indicating some invariance to syntactic structures.

References

[Abadi et al.2015] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

[Arehalli et al.2022] Suhas Arehalli, Brian Dillon, and Tal Linzen. 2022. Syntactic surprisal from neural models predicts, but underestimates, human processing difficulty from syntactic ambiguities. *arXiv preprint arXiv:2210.12187*.

[Ayvaz and Shiha2017] Serkan Ayvaz and Mohammed O. Shiha. 2017. The effects of emoji in sentiment analysis. *International Journal of Computer and Electrical Engineering*, 9:360–369.

[Basile et al.2019] Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63, Minneapolis, Minnesota, USA, June. Association for Computational Linguistics.

[Bassignana et al.2018] Elisa Bassignana, Valerio Basile, and Viviana Patti. 2018. Hurtlex: A multilingual lexicon of words to hurt. In *5th Italian Conference on Computational Linguistics, CLiC-it 2018*, volume 2253, pages 1–6. CEUR-WS.

[Black et al.2021] Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow, March.

[Bojanowski et al.2017] 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.

[Boser et al.1992] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. 1992. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152.

[Brown et al.2020] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020.

Language models are few-shot learners. *CoRR*, abs/2005.14165.

[Cho et al.2014] KyungHyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *CoRR*, abs/1409.1259.

[Chollet and others2015] Francois Chollet et al. 2015. Keras.

[Chowdhery et al.2022] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways.

[Cobbe et al.2021] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.

[Conneau et al.2019] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

[Davidson et al.2017] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media*, volume 11, pages 512–515.

[Devlin et al.2018] 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*.

[ElSherief et al.2021] Mai ElSherief, Caleb Ziems, David Muchlinski, Vaishnavi Anupindi, Jordyn Seybolt, Munmun De Choudhury, and Diyi Yang. 2021. Latent hatred: A benchmark for understanding implicit hate speech. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 345–363, Online and Punta Cana, Dominican Republic, November. Association for Computational Linguistics.

[Epstein et al.2021] Ziv Epstein, Nicolò Foppiani, Sophie Hilgard, Sanjana Sharma, Elena L. Glassman, and David G. Rand. 2021. Do explanations increase the effectiveness of ai-crowd generated fake news warnings? *CoRR*, abs/2112.03450.

[Falcon et al.2019] William Falcon et al. 2019. Pytorch lightning. *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning*, 3.

[Fersini et al.2018a] Elisabetta Fersini, Debora Nozza, and Paolo Rosso. 2018a. Overview of the evalita 2018 task on automatic misogyny identification (ami). *EVALITA Evaluation of NLP and Speech Tools for Italian*, 12:59.

[Fersini et al.2018b] Elisabetta Fersini, Paolo Rosso, and Maria Anzovino. 2018b. Overview of the task on automatic misogyny identification at ibereval 2018. *IberEval@SEPLN*, 2150:214–228.

[Fersini et al.2020] Elisabetta Fersini, Debora Nozza, and Paolo Rosso. 2020. Ami@ evalita2020: Automatic misogyny identification.

[Founta et al.2018] Antigoni Maria 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 *Twelfth International AAAI Conference on Web and Social Media*.

[Grave et al.2018] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

[Guest et al.2021] Ella Guest, Bertie Vidgen, Alexandros Mittos, Nishanth Sastry, Gareth Tyson, and Helen Margetts. 2021. An expert annotated dataset for the detection of online misogyny. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1336–1350, Online, April. Association for Computational Linguistics.

[Hale2001] John Hale. 2001. A probabilistic earley parser as a psycholinguistic model. In *Second meeting of the north american chapter of the association for computational linguistics*.

[Hochreiter and Schmidhuber1997] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

[Huang et al.2022] Fan Huang, Haewoon Kwak, and Jisun An. 2022. Chain of explanation: New prompting method to generate higher quality natural language explanation for implicit hate speech.

[Kumar et al.2018] Ritesh Kumar, Atul Kr Ojha, Shervin Malmasi, and Marcos Zampieri. 2018. Benchmarking aggression identification in social media. In *Proceedings of the first workshop on trolling, aggression and cyberbullying (TRAC-2018)*, pages 1–11.

[LeCun et al.1998] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

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

[Ling et al.2017] Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *CoRR*, abs/1705.04146.

[Liu et al.2017] Haitao Liu, Chunshan Xu, and Junying Liang. 2017. Dependency distance: A new perspective on syntactic patterns in natural languages. *Physics of life reviews*, 21:171–193.

[Liu et al.2019a] Ping Liu, Wen Li, and Liang Zou. 2019a. NULI at SemEval-2019 task 6: Transfer learning for offensive language detection using bidirectional transformers. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 87–91, Minneapolis, Minnesota, USA, June. Association for Computational Linguistics.

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

[Mandl et al.2019] Thomas Mandl, Sandip Modha, Prasenjit Majumder, Daksh Patel, Mohana Dave, Chintak Mandlia, and Aditya Patel. 2019. Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages. In *Proceedings of the 11th forum for information retrieval evaluation*, pages 14–17.

[Mathew et al.2020] Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2020. Hatexplain: A benchmark dataset for explainable hate speech detection. *CoRR*, abs/2012.10289.

[Mikolov et al.2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

[Pennington et al.2014] 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.

[Post2018] Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels, October. Association for Computational Linguistics.

[Radford et al.2019] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

[Raffel et al.2019] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

[Rezvan et al.2018] Mohammadreza Rezvan, Saeedeh Shekarpour, Lakshika Balasuriya, Krishnaprasad Thirunarayan, Valerie L Shalin, and Amit Sheth. 2018. A quality type-aware annotated corpus and lexicon for harassment research. In *Proceedings of the 10th acm conference on web science*, pages 33–36.

[Rumelhart et al.1985] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1985. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.

[Sap et al.2019] Maarten Sap, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A. Smith, and Yejin Choi. 2019. Social bias frames: Reasoning about social and power implications of language. *CoRR*, abs/1911.03891.

[Schmidt and Wiegand2017] Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 1–10, Valencia, Spain, April. Association for Computational Linguistics.

[Seganti et al.2019] Alessandro Seganti, Helena Sobol, Iryna Orlova, Hannam Kim, Jakub Staniszewski, Tymoteusz Krumholc, and Krystian Koziel. 2019.

Nlpr@ srpol at semeval-2019 task 6 and task 5: Linguistically enhanced deep learning offensive sentence classifier. *arXiv preprint arXiv:1904.05152*.

[Sukthanker et al.2020] Rhea Sukthanker, Soujanya Poria, Erik Cambria, and Ramkumar Thirunavukarasu. 2020. Anaphora and coreference resolution: A review. *Information Fusion*, 59:139–162.

[Waseem and Hovy2016a] Zeerak Waseem and Dirk Hovy. 2016a. 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, June. Association for Computational Linguistics.

[Waseem and Hovy2016b] Zeerak Waseem and Dirk Hovy. 2016b. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*, pages 88–93.

[Wei et al.2022] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *CoRR*, abs/2201.11903.

[Zampieri et al.2019a] Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019a. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). *CoRR*, abs/1903.08983.

[Zampieri et al.2019b] Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019b. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). *arXiv preprint arXiv:1903.08983*.

[Zampieri et al.2020] Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. Semeval-2020 task 12: Multilingual offensive language identification in social media (offenseval 2020). *arXiv preprint arXiv:2006.07235*.

[Zhang et al.2022] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models.

[Zhao et al.2021] Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. *CoRR*, abs/2102.09690.

[Zhou et al.2016] Peng Zhou, Zhenyu Qi, Suncong Zheng, Jiaming Xu, Hongyun Bao, and Bo Xu. 2016. Text classification improved by integrating bidirectional lstm with two-dimensional max pooling. *arXiv preprint arXiv:1611.06639*.

A Appendix

	tfidf	countvec	bothvec	ngram_count	ngram_tfidf	ngram_both	pos
emoji_NL w/o #	0.544	0.545	0.543	**0.749**	0.720	0.747	0.746
emoji w/o #	0.520	0.731	0.730	0.738	0.725	0.741	0.744
w/o emoji w/o #	0.530	0.724	0.734	0.738	0.723	0.739	0.744
emoji with #	0.520	0.728	0.729	0.738	0.725	0.741	0.744
emoji_NL with #	0.544	0.545	0.543	**0.749**	0.720	0.747	0.746

Table 6: Test set Macro-F1 scores of Linear SVM on different vectorizers and preprocessing methods testing for emoji and hashtag removal (w/o emoji, w/o #, respectively), and emoji to natural language conversion (emoji_NL).

Model	Hyperparameters	Macro-F1
SVM	c = 0.01, class_weights = balanced	74.9
LSTM	epochs = 50, learning_rate = 1e-4, batch_size = 32, dropout = 0.2	77.23 +_ 1.02
BERT	epochs = 5, learning_rate = 5e-5, batch_size = 16, optimizer = Adam	**80.3 +_ 0.71**
RoBERTa	epochs = 2, learning_rate = 5e-5, batch_size = 32, max_seq_len = 150	75.56 +_ 6.8

Table 7: Macro-F1 scores of baselines and their hyperparameter settings on the test set.

Nr	Prediction	Ground Truth	Explanation	Example
1a	**NOT**	OFF	here you go: .	blank
1b	**NOT**	OFF	here you go: black, black .	duplicates
1c	**NOT**	OFF	here you go: love, people, know, gun, people .	not offensive
2a	**OFF**	NOT	here you go:.	blank
2b	**OFF**	NOT	here you go: red.	not offensive
2c	**OFF**	NOT	here you go: fuck	offensive
2d	**OFF**	NOT	here you go: rape	vague

Table 8: A table of examples for the items mentioned in our error analysis. 'OFF' refers to 'offensive', 'NOT' refers to 'not offensive'. Note that only the explanation side of the prompt is provided.