Can BERT and ChatGPT detect presuppositions of factive predicates?

Anonymous ACL submission

Abstract

Presuppositions—background assumptions assumed to be true by interlocutors—can survive operators that negate other types of inferences such as entailment. This ability makes presuppositions particularly valuable for Natural Language Inference (NLI) tasks. Various lexical and structural elements can trigger these inferences, but their susceptibility to variation and cancellation poses challenges for NLI. Among these triggers, factive predicates are notably variable in their behavior. This paper explores the effectiveness of BERT and ChatGPT-4 trained on the *PresuppositionBank*, a dataset including over 7,500 items featuring factive predicates, in detecting presuppositions triggered by such predicates. Our findings indicate that these models show limited proficiency in this area, highlighting the inherent complexity of the task at hand.

1 Introduction

When a speaker utters a sentence such as (1a), they are not only stating that *Mudd is aware that the president has been assassinated*, but also implying that *the president has been assassinated*. While the first statement is an *Entailment*, the latter is referred to as a *Presupposition*. The difference between an entailment and a presupposition becomes clear when the same sentence is negated. In sentence (1b), we can no longer say that *Mudd is aware that the president has been assassinated*, but the statement that *the president has been assassinated* still holds. This resilience of presuppositions makes them an invaluable resource for Natural Language Processing (NLP) tasks such as text summarization, Question Answering (QA) systems, and Recognizing Textual Entailments (RTE).

(1) a. Mudd *knows* that the president has been assassinated.
b. Mudd <u>does not</u> *know* that the president has been assassinated.

Lexical and structural elements, such as *know*, that lead to presuppositions are known as *presupposition triggers*. In particular, *know* belongs to the category of *factive* predicates which presuppose the truth of the complement clause (CC) that follows them (Kiparsky and Kiparsky, 1970). However, it has been observed that the projection behavior of such predicates can be canceled (Morgan 1969; Langendoen and Savin 1971; Karttunen 1973), among others) or exhibit variability (Tonhauser et al. 2018; De Marneffe et al. 2019; Degen and Tonhauser 2022). For instance, the factive predicate *find out* in (2a) leads to the presupposition that *my mom is sick*. This presupposition does not hold in (2b) when the same sentence is negated. Rather, it can either imply that my mom is sick and I just haven't discovered it yet, or that I didn't find out my mom is sick because she actually isn't.

(2) a. I *found out* that my mom is sick.
b. I <u>didn't</u> *find out* that my mom is sick.

The observed behavior of factive predicates has motivated the creation of the *PresuppositionBank* (Salimifar et al., 2024) for experimental and computational studies on their projectivity. The PresuppositionBank is a collection of over 7,500 items, each featuring a factive predicate under investigation. This paper explores the ability of Large Language Models (LLMs) such as BERT and ChatGPT to detect the projection behavior of factive predicates in the PresuppositionBank. Specifically, the study aims to determine whether BERT (Google, 2019) and ChatGPT (OpenAI, 2024) can accurately predict if the truth of the CC of a factive predicate projects or not. Our findings reveal that BERT tends to over-generalize projectivity and assigns a *Yes* label to most sentences. In contrast, ChatGPT responds randomly, suggesting that it fails to fully comprehend the task.[1]

[1] All data is available in our repository, released under the

2 Related work

Presuppositions can play a significant role in various NLP tasks. Cianflone et al. (2018) focus on adverbial triggers, such as *again*, and identify contexts in which they can be used. They suggest that understanding adverbial presupposition triggers can be useful in summarization tasks and dialogue systems. Kim et al. (2021) demonstrated that detecting the presupposition of a question can enhance the performance of QA systems. Furthermore, several studies have focused on the use of presuppositions in Natural Language Inference (NLI) tasks. An NLI task involves determining the relationship between two sentences. These relationships can be categorized as Entailment (E), Contradiction (C), or Neutral (N).

Jiang and de Marneffe (2019) adapted the CommitmentBank (De Marneffe et al., 2019) for an NLI task. The *CommitmentBank* is a dataset comprising 1,200 items with projectivity judgments collected for 48 clause-embedding predicates. While the original dataset includes scalar certainty ratings for each item (ranging from -3 to +3), Jiang and de Marneffe modified it to suit the requirements of an NLI task. Specifically, they assigned the label E to items with ratings between [1, 3], N to items with a rating of [0], and C to items with ratings between [-3, -1]. They reported that while BERT achieves a high F1 score in predicting the NLI labels, it makes systematic errors.

Jeretic et al. (2020) introduced the *IMPPRESS* diagnostic set to test the ability of NLI models in predicting pragmatic inferences, such as implicature and presupposition. They observed that while BERT can predict the correct label for certain triggers, such as *only*, it does not effectively learn all types of pragmatic inferences. *Nope* (Parrish et al., 2021) is another dataset designed to test the ability of NLI models to detect presuppositions. Parrish et al. reported that these models can make accurate predictions in simple cases. However, similar to previous studies, they stated that models struggle to capture the intrinsic complexities of the task.

Previous studies have primarily examined the capabilities of NLI models on simpler presupposition triggers, which show less projection variability compared to factive predicates. As a result, the latter remains under-explored. Moreover, research that does address factive predicates often relies on limited or artificially constructed datasets (De Marneffe et al., 2019; White and Rawlins, 2018). These constraints adversely affect the representativeness of computational tasks as they fail to capture the inherent variability in the projectivity of factive predicates.

To address this gap, the *PresuppositionBank* (Salimifar et al., 2024) has been compiled. This dataset focuses on the projectivity of certain factive predicates, specifically those that exhibit the highest variability in their projection behavior. While the selected predicates are all considered factive, their variable projection behavior has led to ascribing them to different categories. Consequently, the variability we refer to includes both within-category and between-category differences.[2] For instance, although predicates *acknowledge* and *admit* are both verbs of communication, experimental studies have demonstrated that *acknowledge* tends to be more projective than *admit* (Degen and Tonhauser, 2022; Salimifar, 2023).

3 Presupposition detection

As observed in the previous section, an inferencing task involves selecting one of the three labels—E, N, or C—for the relationship between two sentences, the *premise* and the *hypothesis*. Consequently, most NLI datasets are collected by having participants choose one of these labels for the relationship between two sentences, or by providing a premise and asking participants to write three sentences that correspond to the E, C, and N relations to the premise (Bowman et al. 2015; Williams et al. 2018). Other datasets, such as the CommitmentBank, are later adapted for this task by mapping their labels to one of these three categories. An instance of the mapped labels according to Jiang and de Marneffe (2019) for the CommitmentBank is provided in Figure 1.

While the E and N labels in Figure 1 are straightforward, the third example, which is mapped to a C label, is more controversial. When collecting the original dataset, De Marneffe et al. (2019) gathered participants' responses to the "certain that" diagnostic test. This diagnostic test specifically asks how certain the speaker is about the CC of the clause-embedding predicate in question. For instance, the "certain that" diagnostic test for the

Attribution-ShareAlike 4.0 International license. The codes employ libraries from HuggingFace (HuggingFace, 2020), which are licensed under Apache 2.0.

[2] For more information on the proposed categories of factive predicates in the literature: (Karttunen, 1973), (Karttunen, 2016), (Salimifar, 2023), among others.

C label in Figure 1 likely involved a prompt such as *Tell us how certain the speaker is that Japan is opening up to more foreign takeovers*. Therefore, one cannot equate a certainty rating of -1.92, which roughly translates to *It is not certain that Japan is opening up to more foreign takeovers*, with a contradiction. If anything, it should be considered a neutral relation. As a result, it is not sensible to define the current task as a conventional NLI task aimed at finding E, N, and C labels for factive predicates, as a true C label may not exist. Consequently, we define the task at hand as detecting presuppositions triggered by factive predicates in sentences. More specifically, predicting *Yes*, if the CC of the predicate projects, or *No* if it does not.

Additionally, we noted that NLI models have a limited ability to make inferences based on presuppositions, a limitation evident in both complex and simpler cases. For instance, Kabbara and Cheung (2023) investigated whether NLI models truly learn to make pragmatic inferences or merely rely on factors like lexical overlap. They used the presupposition part of the IMPRESS dataset (Jeretic et al., 2020), which includes simpler presupposition triggers such as cleft constructions, questions, and possessive definites. Despite these cases being less lexicalized or dependent on world knowledge—as opposed to factive predicates— the models still struggled to make inferences based on pragmatic reasoning. Basmov et al. (2023) reported similar performance for large language models like ChatGPT, noting that even in cases trivial for humans, these models often fail to make the correct inference.

These observations highlight the complexity of the current task, as detecting the projectivity of factive predicates is significantly more challenging. This complexity arises from the nature of factive predicates, which—despite their overall high projectivity—exhibit variability in their projection behavior. Subsequently, this leads to an unbalanced dataset where the *Yes* labels predominate. Therefore, capturing their behavior in specific cases where they don't project is not an easy task. However, given their ubiquity in everyday communication and their potential to enhance language understanding, it is a task worth pursuing.

3.1 Dataset splits

The data was divided into training, development, and testing sets for the machine learning training. Salimifar et al. (2024) applied several statistical models to the PresuppositionBank and found that the model predicting projectivity based on the interaction between predicate lemmas and the presence or absence of an entailment-cancelling operator (ECO) explained more variability than other models. To examine the impact of ECO on the projectivity of factive predicates, we tested our model on three versions of the dataset: 1) including both sentences with and without ECO, 2) including only sentences with ECO, and 3) including only sentences without ECO.

Additionally, to test the potential influence of context on the final predicted labels, the original dataset comprised two distinct versions: one with context and one without context. Therefore, we ensured that sentences appearing in the training set of the with-context dataset also appeared in the training set of the without-context dataset, maintaining consistency across versions. The dataset splits followed the commonly applied ratios of 0.8/0.1/0.1 for training, development, and testing sets, respectively. Table 1 represents the number of sentences with *Yes* and *No* final labels in each split.

Furthermore, we analyzed the data to balance two different properties across the splits: (1) the number of sentences labeled with aggregated *Yes* and *No* responses, and (2) the number of sentences in which the predicate is embedded within an entailment-cancelling operator, as not all sentences are embedded. To achieve this, we first counted the number of sentences corresponding to each property. Subsequently, we calculated the number of sentences for each property that needed to be assigned to each split based on the determined ratios. Finally, we randomly shuffled the dataset and assigned the sentences to the training, develop-

Premise: A: Boy that's scary, isn't it. B: Oh, can you imagine, because it happens in the middle of the night, so you know, these parents didn't know the kid was gone until the kid is knocking on the door screaming, let me in.
Hypothesis: the kid was gone. `Entailment (1.92)`

Premise: "Clever". Klug means "clever". Would you say that Abie was clever?
Hypothesis: Abie was clever. `Neutral (0)`

Premise: GM confirmed it received U.S. antitrust clearance to boost its holding. Sansui Electric agreed to sell a 51% stake to Polly Peck of Britain for $110 million. Still, analysts said the accord doesn't suggest Japan is opening up to more foreign takeovers.
Hypothesis: Japan is opening up to more foreign takeovers. `Contradiction (-1.2)`

Figure 1: Examples of NLI mapped labels from the CommitmentBank.

Dataset	*Yes*	*No*
Train	5023	1032
Test_with_ECO	274	48
Test_without_ECO	354	81
Test_all	628	129
Dev	630	130

Table 1: Number of sentences with *Yes* and *No* labels in each dataset.

ment, and testing sets in a manner that ensured the properties were balanced across the splits.

3.2 Feature vector extraction

To evaluate the influence of different layers and various token types in sentences on predicting the final *Yes* or *No* label, we extracted feature vectors for the CC of the factive predicate in question, the entailment-cancelling operator (ECO), the factive predicate (P), and the special [CLS] token at the beginning of each sentence. This process was carried out for both with-context and without-context versions of the dataset.

We used the BERT-base-cased model, employing PyTorch implementation of the model from HuggingFace, which consists of 12 hidden layers and an initial layer for input embeddings. During the experiments, all hidden layers were frozen. The model and the torch were set to evaluation and no_grads (no gradient evaluation) modes, respectively. After inputting the sentences into the model, we obtained token and layer outputs for each layer and every token in the sentences. When there were multiple tokens of the same type, such as ECO and P, their vectors were summed to produce a single vector for each type. For CC tokens, we used the torch.max function to combine their vectors, as this approach provided better results. For each sentence and each layer, we extracted four vectors: [CLS], CC, ECO, and P, each with a size of 768 (the default BERT token size). To aggregate information across different layers in multi-layer configurations, we summed the vectors obtained from multiple layers.

3.3 Training

We employed a torch.nn.Linear (PyTorch, 2023c) linear classifier, which is frequently used in classification tasks (HuggingFace, 2020). The classifier was initialized with varying input sizes determined by the token configuration, with two output classes (*Yes* and *No*). We concatenated the token vectors, applied a dropout layer with a dropout ratio of 0.1, and subsequently fed them into the classifier. Given the binary nature of the classification task, we utilized the CrossEntropyLoss function provided by PyTorch (PyTorch, 2023b). The hyper-parameters were as follows: a learning rate of 0.00002 with a linear scheduler and no warm-up steps, the AdamW optimizer (PyTorch, 2023a), a batch size of 8, and a total of 10 epochs.

Each training was conducted five times with different random seeds. Additionally, to mitigate the variations caused by initialization, the mean score of these runs was computed. The model obtaining the highest F1 score on the development split at the end of each epoch was selected as the best model. The results for other metrics were reported based on this model. During a single training session, a total of 1,967,280 parameters were trained (with varying sizes of linear layers depending on the experiment), and the process took approximately 59 minutes on a lightweight compute node equipped with an NVIDIA GeForce RTX 2080 GPU, provided by our university's computing infrastructure for research purposes.

4 Results

Table 2 represents the highest mean accuracy for each test set along with their corresponding layers and configurations. Due to the lack of a prior baseline for this task and the general imbalance in the number of sentences labeled with *Yes* versus *No* responses in the original dataset, we defined the baseline as the scenario where all sentences are assumed to have a final *Yes* label. It is evident that the model's best performance exceeds the baseline by only about 1 percent.[3] A closer look at the number of failed *Yes* and *No* responses in Table (3) reveals that the models generally struggle with predicting the *No* labels.

Additionally, no significant difference can be observed in the model's performance between the with_ECO and without_ECO test sets. However, all configurations that achieve better results incorporate feature vectors related to both ECO and P, supporting the findings of Salimifar et al. (2024) that the interaction between entailment-canceling operators and predicate lemmas is effective. Conversely, the impact of context appears random, as it does not consistently enhance performance.

[3] A comprehensive list of all the obtained results from the five runs is provided on the repository.

Test_set	Layer	Config	Baseline	Ave_Acc	Δ
Test	10	ECO_P_without_context	0.9068	0.9155	0.0086
Test_with_ECO	9-10-11-12	ECO_P_CLS_without_context	0.9194	0.9313	0.0118
Test_without_ECO	11	ECO_P_context	0.8973	0.9095	0.0121

Table 2: BERT performance on the three versions of the test set. Ave_Acc indicates the average accuracy over five runs, and Δ represents the improvement of the Ave_Acc over the baseline.

Test_set	Layer	Config	All_No	Failed_No	All_Yes	Failed_Yes
Test	10	ECO_P_without_context	129	103	628	8
Test_with_ECO	9-10-11-12	ECO_P_CLS_without_context	48	34	274	3
Test_without_ECO	11	ECO_P_context	81	66	354	1

Table 3: Number of failed predictions for *Yes* and *No* labels for the best models.

5 Discussion

ChatGPT Given the celebrated capabilities of large language models (LLMs) like ChatGPT, we aimed to assess their proficiency in detecting the presuppositions triggered by factive predicates. Specifically, we used ChatGPT-4 (OpenAI, 2024), providing it with our training set, explaining the task, and requesting it to assign a final *Yes* or *No* label to sentences in the test set.[4] However, the labels provided by ChatGPT were observed to be random and unreliable. In particular, we repeated the experiment with ChatGPT five times, as we had done with BERT. The results varied significantly between runs, with one run producing mostly *Yes* responses and another mostly *No* responses. This variability indicates that LLMs like ChatGPT also struggle with this task.

General discussion The observed behavior from BERT and ChatGPT demonstrated that such models fail to predict the projective behavior of factive predicates. This is not unexpected as previous literature on simpler cases, including triggers that are less variable in their projectivity, also revealed that such models are limited in their abilities and are not able to make pragmatic inferences for the right reasons. Furthermore, predicting the projective behavior of factive predicates is challenging for humans as well. Salimifar et al. (2024) reported that more than 50% of the sentences in the PresuppositionBank received either two or three *Yes* ratings out of five, indicating a higher within-annotator variability for over half of the dataset. This suggests that determining the projectivity of factive predicates is difficult not only for language models but also for human annotators.

Another challenge for the current task arises from an unbalanced dataset. While factive predicates exhibit variability in their projective behavior, they are predominantly projective—as shown in Table (1). Therefore, the number of final *Yes* responses greatly exceeds the *No* responses. One solution is to increase the number of items in the dataset beyond those currently available in the PresuppositionBank, providing the model with more data and greater variability to learn from, which could consequently enhance its performance.

Additionally, although context has not shown a significant influence on the results at this stage, we plan to investigate its impact in future stages using minimal pairs. Specifically, we will present the same sentences with and without preceding discourse to different participants to determine if context plays a role. Finally, given the observed influence of entailment-cancelling operators, we aim to replicate a similar experiment with context for embeddings to examine their influence in more detail.

6 Conclusion

This study investigated the ability of BERT and ChatGPT to predict the projectivity of factive predicates in the PresuppositionBank. The findings demonstrated that both BERT and ChatGPT struggle to make systematic predictions, highlighting the complexity of the task. Additionally, while context did not significantly influence the results, predicate lemma and entailment-cancelling operators appeared to improve BERT's accuracy in predicting the final label.

[4]The prompts used for this experiment are provided in the Appendix.

7 Limitations

One limitation of this work, as with most statistical or neural approaches to NLP tasks, is the requirement for large datasets for efficient training, with such data being predominantly available in English. Additionally, this paper directly tackles the limitations of LLM models, such as BERT and ChatGPT, by examining the boundaries of their capabilities, particularly in understanding the presuppositions of factive predicates. While these models excel in some areas, they continue to face challenges in tasks that demand deep contextual comprehension and reasoning.

References

Victoria Basmov, Yoav Goldberg, and Reut Tsarfaty. 2023. Chatgpt and simple linguistic inferences: Blind spots and blinds. *arXiv preprint arXiv:2305.14785*.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Andre Cianflone, Yulan Feng, Jad Kabbara, and Jackie Chi Kit Cheung. 2018. Let's do it "again": A first computational approach to detecting adverbial presupposition triggers. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2747–2755, Melbourne, Australia. Association for Computational Linguistics.

Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The CommitmentBank. In *SuB 23*, pages 107–124.

Judith Degen and Judith Tonhauser. 2022. Are there factive predicates? *Language*, 98:552–591.

Google. 2019. google-bert/bert-base-cased. https://huggingface.co/google-bert/bert-base-cased, Accessed: 25.06.2024.

HuggingFace. 2020. Modeling bert. https://github.com/huggingface/transformers/blob/2daf2c3eaa2992018b8d1ceb3421f5baa649bad5/src/transformers/models/bert/modeling_bert.py#L1661, Accessed: 26.06.2024.

Paloma Jeretic, Alex Warstadt, Suvrat Bhooshan, and Adina Williams. 2020. Are natural language inference models IMPPRESsive? Learning IMPlicature and PRESupposition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8690–8705, Online. Association for Computational Linguistics.

Nanjiang Jiang and Marie-Catherine de Marneffe. 2019. Evaluating bert for natural language inference: A case study on the commitmentbank. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 6086–6091.

Jad Kabbara and Jackie Cheung. 2023. Investigating the effect of pre-finetuning BERT models on NLI involving presuppositions. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10482–10494, Singapore. Association for Computational Linguistics.

Lauri Karttunen. 1973. Presuppositions of compound sentences. *Linguistic inquiry*, 4(2):169–193.

Lauri Karttunen. 2016. Presupposition: What went wrong? In *SALT 26*, pages 705–731.

Najoung Kim, Ellie Pavlick, Burcu Karagol Ayan, and Deepak Ramachandran. 2021. Which linguist invented the lightbulb? presupposition verification for question-answering. *arXiv preprint arXiv:2101.00391*.

Paul Kiparsky and Carol Kiparsky. 1970. Fact. In Manfred Bierwisch and Klaus E. Heidolph, editors, *Progress in Linguistics: A Collection of Papers*, pages 143–173. Mouton, The Hague.

D. Terence Langendoen and Harris B. Savin. 1971. The projection problem for presuppositions. In Charles J. Fillmore and D. Terence Langendoen, editors, *Studies in Linguistic Semantics*, pages 54–60. Holt, Rinehart & Winston, New York.

Jerry L Morgan. 1969. On the treatment of presupposition in transformational grammar. In *Proceedings from the Annual Meeting of the Chicago Linguistic Society*, volume 5, pages 167–177. Chicago Linguistic Society.

OpenAI. 2024. Chatgpt conversation on june 28, 2024. Accessed: 2024-06-28.

Alicia Parrish, Sebastian Schuster, Alex Warstadt, Omar Agha, Soo-Hwan Lee, Zhuoye Zhao, Samuel R. Bowman, and Tal Linzen. 2021. NOPE: A corpus of naturally-occurring presuppositions in English. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 349–366, Online. Association for Computational Linguistics.

PyTorch. 2023a. AdamW - PyTorch 2.3 documentation. https://pytorch.org/docs/stable/generated/torch.optim.AdamW.html, Accessed: 27.06.2024.

PyTorch. 2023b. CrossEntropyLoss - PyTorch 2.3 documentation. https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html, Accessed: 26.06.2024.

PyTorch. 2023c. Linear - PyTorch 2.3 documentation. https://pytorch.org/docs/stable/generated/torch.nn.Linear.html, Accessed: 26.06.2024.

S. Salimifar, D.T.T. Haug, and Ahmet Yildirim. 2024. The Presuppositionbank: a dataset on presupposition projection of factive predicates. Unpublished manuscript.

Saeedeh Salimifar. 2023. Presupposition projection: Let's test again! In *ConSOLE XXXI*, pages 93–108.

Judith Tonhauser, David I Beaver, and Judith Degen. 2018. How Projective is Projective Content? *Journal of Semantics*, 35:495–542.

Aaron Steven White and Kyle Rawlins. 2018. The role of veridicality and factivity in clause selection. In *NELS 48*, pages 221–234.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

A Appendix

Prompt used for ChatGPT

I will provide you with a 3-column file. The first column contains sentences, the second column contains questions related to those sentences, and the third column contains the corresponding answers. I would like you to analyze this file to understand its structure and the labels used.
Afterward, I will give you a different file that only includes two columns: the sentences and the questions. Your task will be to provide answers to the questions. You should respond with either "Yes, it is" or "No, it isn't".

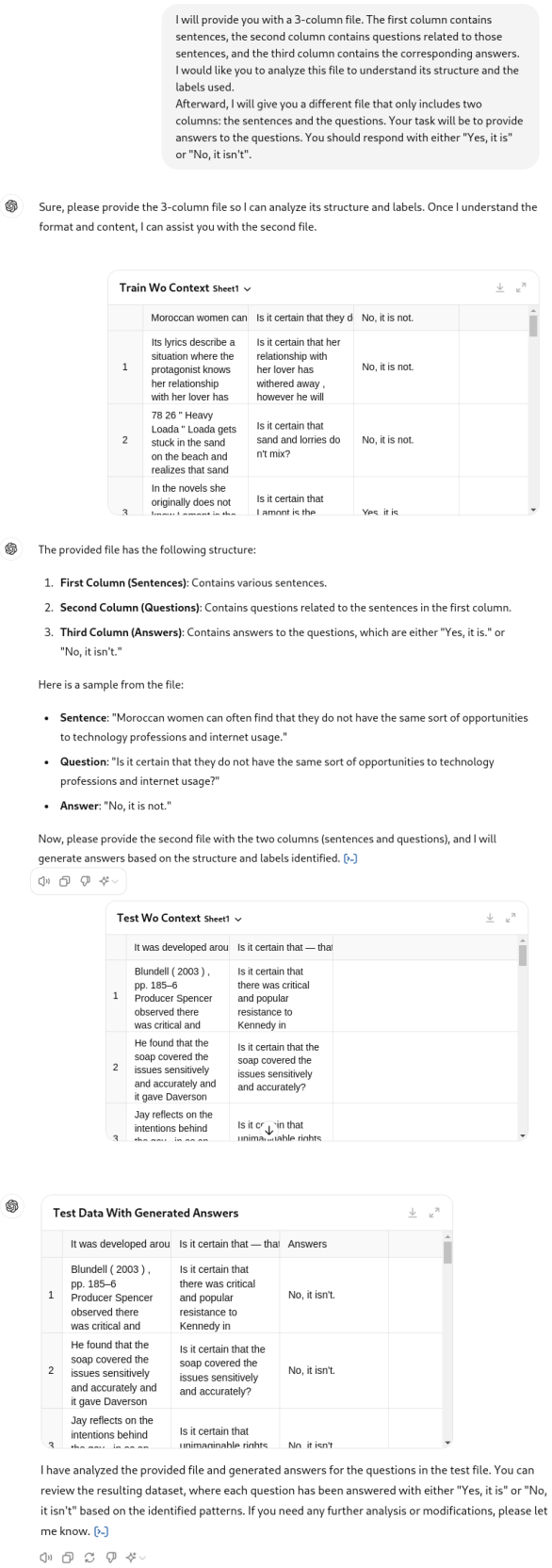

Figure 2: Prompt used with ChatGPT.