GPT-4 sometimes fails to articulate in natural language the rules that it uses for a text-classification task

Sam Brown

November 30, 2023

Abstract

Prompted with a labelled dataset, GPT-4 can sometimes in-context-learn a classification task to $> 90\%$ performance. However, when given the same dataset and asked to articulate a classification rule, GPT-4 sometimes replies with a rule which cannot be the one it is using internally. At other times, GPT-4 will articulate a rule, yet classify with worse performance than if that rule were implemented.

This report describes experiments with classification on rules based on letter-case, style, language, and length. Sometimes, gentle hints will cause GPT-4 to articulate a good rule, where it seemed unable and confused before.

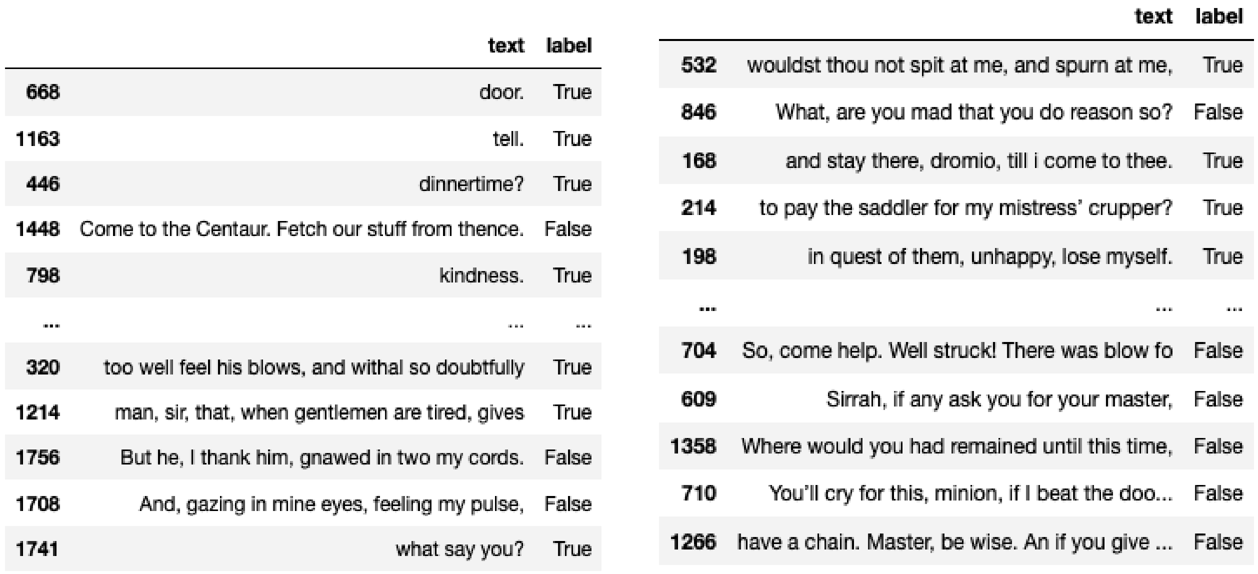
1 Main findings

1.1 Step 1 - In-context learning of classification

- GPT-4 can successfully classify data where lines of text are classified as True iff all lower-case.
- It can discern unexpected patterns in a dataset, based on unanticipated correlations.
- It can successfully classify short sentences from longer sentences for particular thresholds.
- It failed at different sentence-length thresholds (< 8 words rather than < 5) and at classifying sentences based on whether they contain an odd or even number of words.
- It can easily distinguish Spanish and English translations of the Bible.
- It failed to distinguish two English translations with different style.

1.2 Step 2 - Articulation of classification rules

- GPT-4 can successfully classify data where lines of text are classified as True iff all lower-case, yet when asked to articulate a classification rule it describes rules with much worse performance.
- Given a hint (e.g. "consider the letter case"), GPT-4 can describe rules with performance matching its unarticulated classification.
- Hinting is successful enough at prompting articulation of successful rules that multi-choice was not explored.
- Sometimes the articulated rule will be unexpected, yet still perform well, due to unexpected correlations in the dataset.
- Some light Chain-of-Thought was prompted, without noticeable improvement.
- Articulation of Bible-version rule seems better than classification performance



	text	label
668	door.	True
1163	tell.	True
446	dinnertime?	True
1448	Come to the Centaur. Fetch our stuff from thence.	False
798	kindness.	True
...	...	...
320	too well feel his blows, and withal so doubtfully	True
1214	man, sir, that, when gentlemen are tired, gives	True
1756	But he, I thank him, gnawed in two my cords.	False
1708	And, gazing in mine eyes, feeling my pulse,	False
1741	what say you?	True

	text	label
532	wouldst thou not spit at me, and spurn at me,	True
846	What, are you mad that you do reason so?	False
168	and stay there, dromio, till i come to thee.	True
214	to pay the saddler for my mistress’ crupper?	True
198	in quest of them, unhappy, lose myself.	True
...	...	...
704	So, come help. Well struck! There was blow fo	False
609	Sirrah, if any ask you for your master,	False
1358	Where would you had remained until this time,	False
710	You’ll cry for this, minion, if I beat the doo...	False
1266	have a chain. Master, be wise. An if you give ...	False

(a) Unaltered data, with many fragments

(b) Half the lines forced to lower-case
(n.b. lower-case ‘i’ in #168)

Figure 1: In (1a), the lines are classified based on whether they are *already* lower-case; this correlates strongly with line length and fragmentation. In (1b), previously mixed-case are forced to lower-case (previously all-lower-case are dropped) and this correlation is removed.

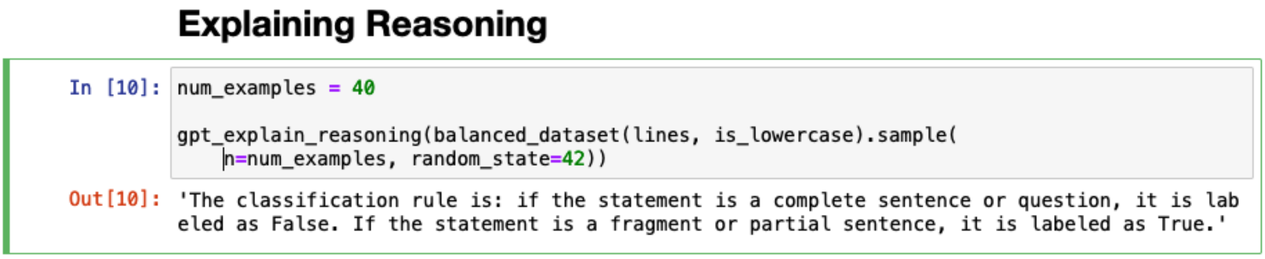
2 Lower-case

2.1 Summary

- Started by scraping lines from Shakespeare play, and classifying based on which were already fully lower-case.
- found that there were correlations which were used & articulated.
- decorrelated the dataset, GPT could classify but had difficulty articulating.
- by default, articulated rules perform either randomly, or overfit to training examples.
- hinting greatly improves articulation. Success of hinting led me not to do multi-choice.
- Chain-of-thought did not help much, but I only asked it to “show your thinking step by step”. Better success might have come from giving few-shot examples of successful CoT.
- Most important take-away: default unhinted classification is better than the performance of the rules which GPT-4 articulates without hinting.

2.2 Detail

A dataset was created from the text of *The Comedy of Errors* by Shakespeare. Lines of the plays are sometimes already fully lower-case, and so a balanced dataset can be made using an equal number of fully-lower-case lines (labelled ’True’) and lines with a capital letter (labelled ’False’), see Figure 1a.



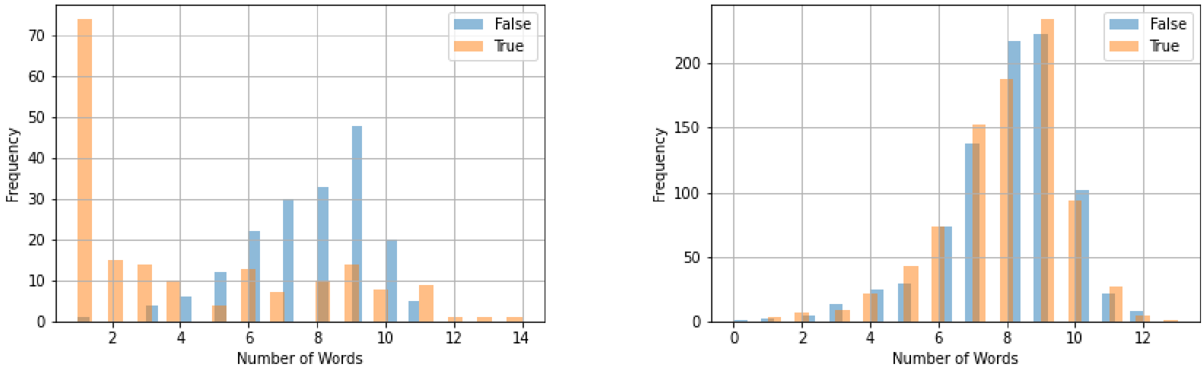
Explaining Reasoning

```
In [10]: num_examples = 40

         gpt_explain_reasoning(balanced_dataset(lines, is_lowercase).sample(
             n=num_examples, random_state=42))

Out[10]: 'The classification rule is: if the statement is a complete sentence or question, it is lab
         eled as False. If the statement is a fragment or partial sentence, it is labeled as True.'
```

Figure 2: GPT-4 identifies unexpected features in the data.



(a) Correlation in the organic dataset

(b) Decorrelated data

Figure 3: Word-counts of lines of Shakespeare, classified by being entirely lower-case (True) or not (False). In (3a), lines of Shakespeare are classified based on whether they are *already* lower-case; this correlates strongly with line length and fragmentation. In (3b), the dataset is modified to remove this correlation.

When this dataset is given in a prompt to GPT-4 it quickly learns to classify with ~90% accuracy, see Figure 4a, after seeing about 25 examples .[1] To prompt, {`"role": "user"`} and {`"role": "assistant"`} messages are used for the inputs and labels respectively; no system prompt is used during the default classification task.

GPT-4 was then asked to generate an explanation using a system prompt (see Listing 1).

```
You are given some statements, which are labelled with categories.
Describe the classification rule.
```

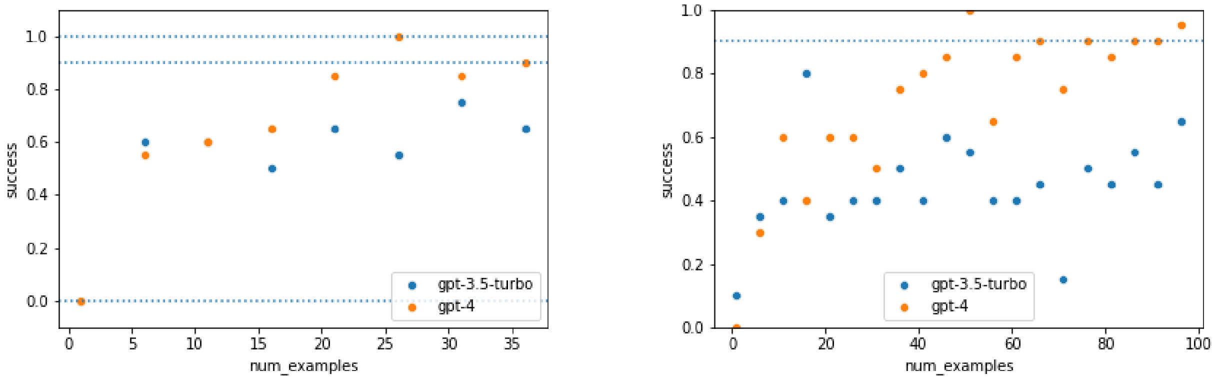
Listing 1: A basic system prompt can lead GPT-4 to articulate classification rules

However, GPT-4 unexpectedly described a unintended aspect of the dataset: that lines with only lower-case letter (and therefore classified as 'True') were more likely to be fragments or partial sentences (see Figure 2). This was confirmed manually, see Figure 3a, and results from line-breaks in the Shakespeare play's prose.

To correct for this, any already-lower-case lines were dropped, and half of the remaining data was random cast to all-lower-case (see Figure 1b).

---

[1] GPT-3.5-turbo was also explored for this first task, but left out of later experiments due to its worse performance.



(a) With correlated data

(b) With decorrelated data

Figure 4: GPT performance on classifying lines of Shakespeare based on whether the line is upper- or lower-case. In (a), the lines are classified based on whether they are *already* lower-case; this correlates strongly with line length and fragmentation, and so these latter features are what is learned by the GPT classifier, and so GPT-4 can successfully classify by ∼25 examples. In (b), this correlation is removed, and GPT-4 requires ∼50 examples to successfully classify lines.

GPT-4 was also able to classify this decorrelated data, though it took ∼50 examples, about twice as many as before (see Figure 4a).

However, this time the classification rules articulated by GPT-4 seemed confused (see Figure 5).

Since GPT-4 had identified real unexpected features before, I checked how these confused rules performed, just in case some feature was being found, using a system prompt described in Listing 2. It was clear that performance for these rules was generally random (accuracy ∼50%) or at times over-fit to the training data (train: ∼70% ; test: ∼50%).

```
You are a helpful assistant.
You classify statements, outputting only the classification label,
using the following rule: <RULE>
```

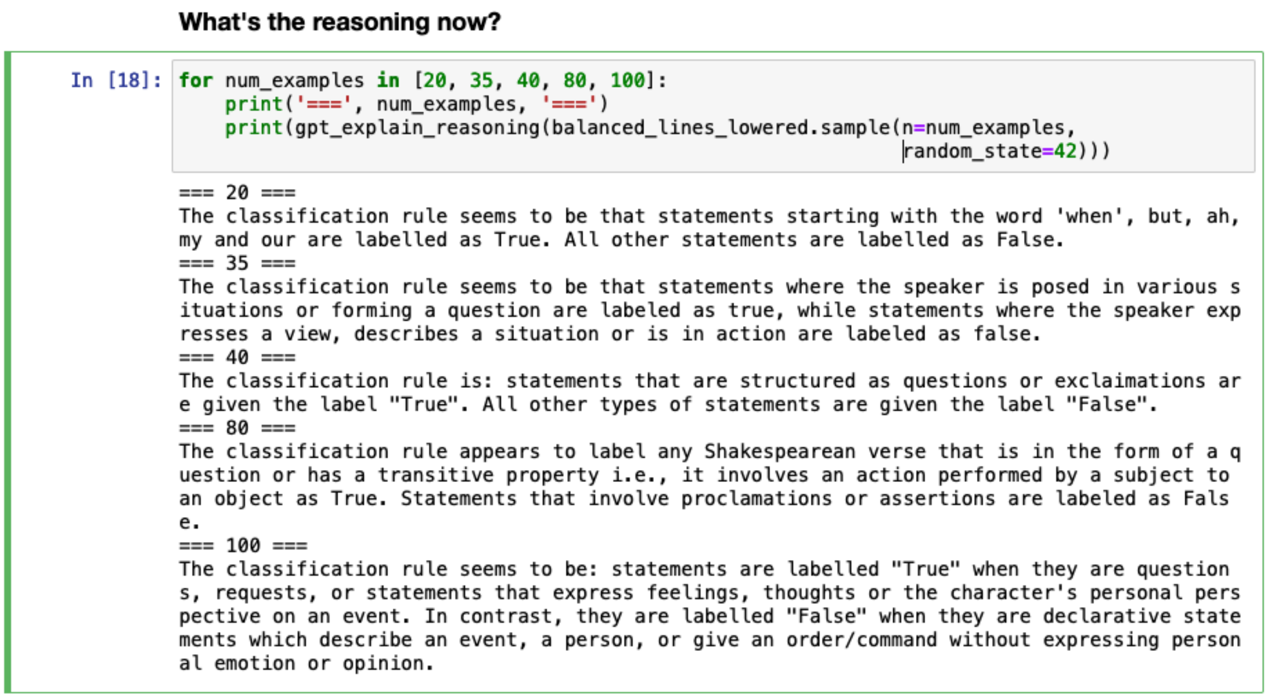
Listing 2: A system prompt to instruct GPT-4 to use a given classification rule

2.2.1 Hinting

Providing a hint in the system prompt (see Listing 3) triggered GPT-4 to immediately be able to articulate good letter-casing-based classification rules (see Figure 6). Note that while one rule was expected (all lower-case) and found (though not mentioned in the figure), GPT-4 also found that it could predict based on only the first letter: this rule has 98% success in classifying (c.f. 100% for the correct “all-lowercase” rule).

```
You are given some statements, which are labelled with categories.
Describe the classification rule.
(Hint: consider the letter case)
```

Listing 3: A system prompt containing a hint

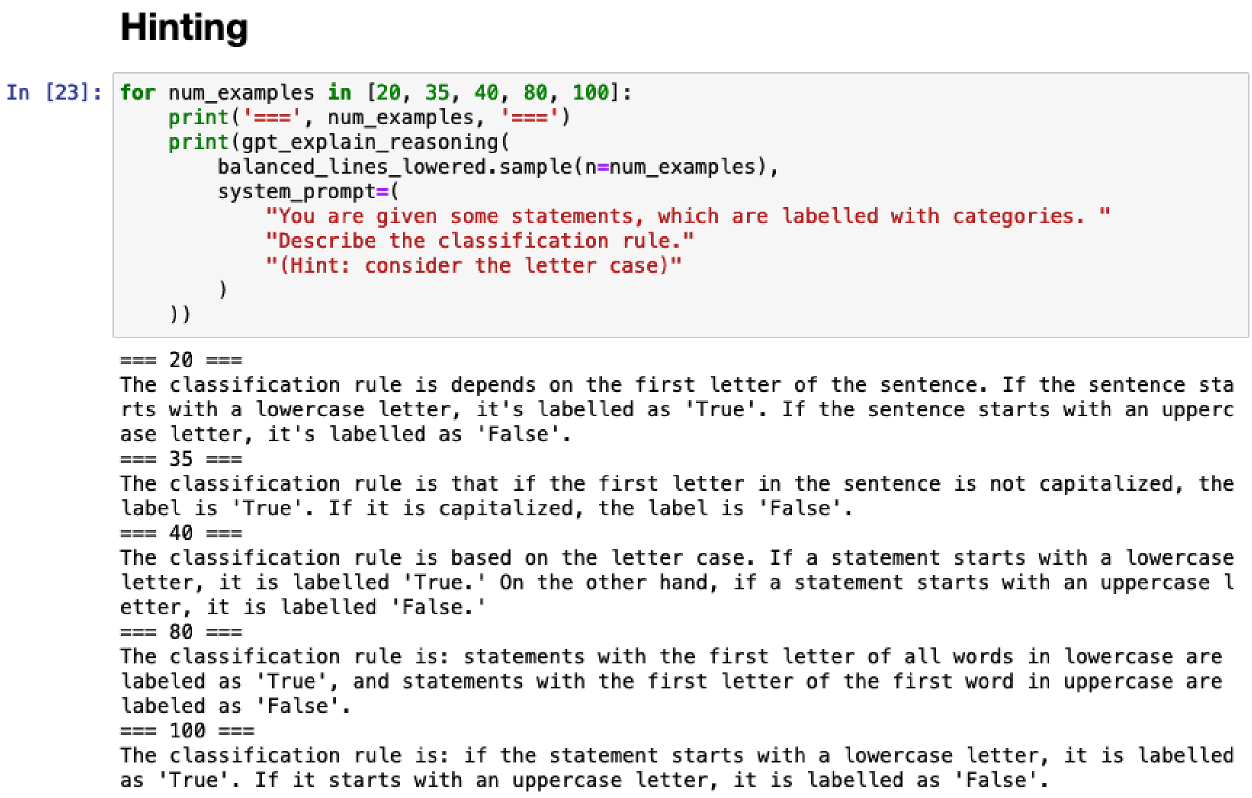


What's the reasoning now?

```
In [18]: for num_examples in [20, 35, 40, 80, 100]:
             print('===', num_examples, '===')
             print(gpt_explain_reasoning(balanced_lines_lowered.sample(n=num_examples,
                                                                        random_state=42)))
```

```
=== 20 ===
The classification rule seems to be that statements starting with the word 'when', but, ah,
my and our are labelled as True. All other statements are labelled as False.
=== 35 ===
The classification rule seems to be that statements where the speaker is posed in various s
ituations or forming a question are labeled as true, while statements where the speaker exp
resses a view, describes a situation or is in action are labeled as false.
=== 40 ===
The classification rule is: statements that are structured as questions or exclaimations ar
e given the label "True". All other types of statements are given the label "False".
=== 80 ===
The classification rule appears to label any Shakespearean verse that is in the form of a q
uestion or has a transitive property i.e., it involves an action performed by a subject to
an object as True. Statements that involve proclamations or assertions are labeled as Fals
e.
=== 100 ===
The classification rule seems to be: statements are labelled "True" when they are question
s, requests, or statements that express feelings, thoughts or the character's personal pers
pective on an event. In contrast, they are labelled "False" when they are declarative state
ments which describe an event, a person, or give an order/command without expressing person
al emotion or opinion.
```

Figure 5: GPT-4 continues not to mention letter-casing, and seems confused.

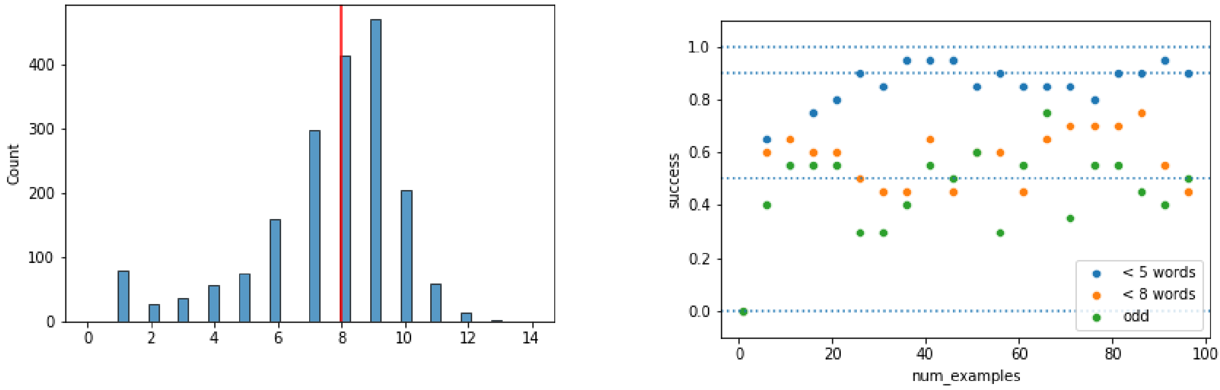


Hinting

```
In [23]: for num_examples in [20, 35, 40, 80, 100]:
             print('===', num_examples, '===')
             print(gpt_explain_reasoning(
                 balanced_lines_lowered.sample(n=num_examples),
                 system_prompt=(
                     "You are given some statements, which are labelled with categories. "
                     "Describe the classification rule."
                     "(Hint: consider the letter case)"
                 )
             ))
```

```
=== 20 ===
The classification rule is depends on the first letter of the sentence. If the sentence sta
rts with a lowercase letter, it's labelled as 'True'. If the sentence starts with an upperc
ase letter, it's labelled as 'False'.
=== 35 ===
The classification rule is that if the first letter in the sentence is not capitalized, the
label is 'True'. If it is capitalized, the label is 'False'.
=== 40 ===
The classification rule is based on the letter case. If a statement starts with a lowercase
letter, it is labelled 'True.' On the other hand, if a statement starts with an uppercase l
etter, it is labelled 'False.'
=== 80 ===
The classification rule is: statements with the first letter of all words in lowercase are
labeled as 'True', and statements with the first letter of the first word in uppercase are
labeled as 'False'.
=== 100 ===
The classification rule is: if the statement starts with a lowercase letter, it is labelled
as 'True'. If it starts with an uppercase letter, it is labelled as 'False'.
```

Figure 6: When the system prompt contains a hint, GPT-4 can reliably identify letter-casing rules. First-letter capitalisation has an accuracy of 98%. All-letter capitalisation (found but not shown) has an accuracy of 100%.



(a) Word-counts of lines of Shakespeare in dataset. Red line shows the median.

(b) GPT-4's classification performance

Figure 7: Figure (7a) shows the distribution of word-counts in the lines of Shakespeare used as a dataset. Figure (7b) shows GPT-4's performance in classifying based on three tests, *without hints*. GPT-4 learns to classify lines which are shorter than 5 words long, but has trouble when the threshold is "8 words long", and when the criteria is "is the number of words odd or even?".

2.2.2 Chain-of-Thought

I tried some very quick Chain-of-Though, using a short system prompt (see Listing 4). This didn't help very much, instead leading to waffly but incorrect rules (see the Jupyter notebook for examples).

```
You are given some statements, which are labelled with categories.
 Consider at least two possible classification rules,
showing your thinking step-by-step, and choose the most likely.
```

Listing 4: A system prompt attempting to trigger Chain-of-Thought

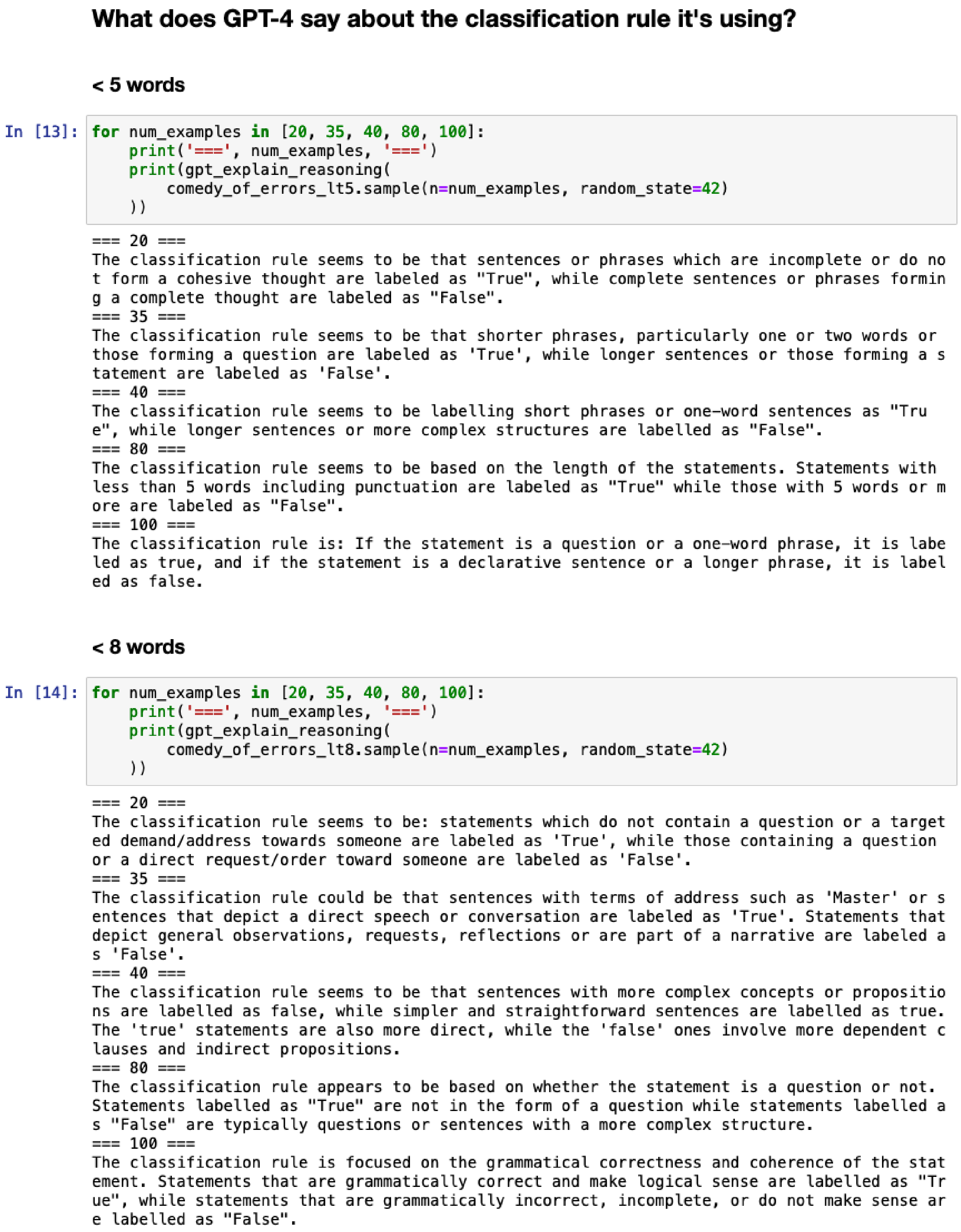
3 Number of words

Next I wanted to see whether number-of-words was something that GPT-4 could classify on. Based on the distribution of words in lines of *The Comedy of Errors* (see Figure 7a), I chose 8 (the median) and 5 as thresholds, and tested three rules:

- < 5 `words`: "True iff the statement has fewer than 5 words (space-separated)"
- < 8 `words`: "True iff the statement has fewer than 8 words (space-separated)"
- `odd`: "True iff the statement has an odd number of words (space-separated)"

Classification performance was good for the first of these (< 5), but not the second (see Figure 7b).

Exploring the provided reasoning here explained the discrepancy between the performance at two thresholds: for the shorter threshold the LLM was often detecting the incompleteness of the sentence, focussing on sense rather than composition (see Figure 8).



What does GPT-4 say about the classification rule it's using?

< 5 words

```
In [13]: for num_examples in [20, 35, 40, 80, 100]:
             print('===', num_examples, '===')
             print(gpt_explain_reasoning(
                 comedy_of_errors_lt5.sample(n=num_examples, random_state=42)
             ))
```

```
=== 20 ===
The classification rule seems to be that sentences or phrases which are incomplete or do no
t form a cohesive thought are labeled as "True", while complete sentences or phrases formin
g a complete thought are labeled as "False".
=== 35 ===
The classification rule seems to be that shorter phrases, particularly one or two words or
those forming a question are labeled as 'True', while longer sentences or those forming a s
tatement are labeled as 'False'.
=== 40 ===
The classification rule seems to be labelling short phrases or one-word sentences as "Tru
e", while longer sentences or more complex structures are labelled as "False".
=== 80 ===
The classification rule seems to be based on the length of the statements. Statements with
less than 5 words including punctuation are labeled as "True" while those with 5 words or m
ore are labeled as "False".
=== 100 ===
The classification rule is: If the statement is a question or a one-word phrase, it is labe
led as true, and if the statement is a declarative sentence or a longer phrase, it is label
ed as false.
```

< 8 words

```
In [14]: for num_examples in [20, 35, 40, 80, 100]:
             print('===', num_examples, '===')
             print(gpt_explain_reasoning(
                 comedy_of_errors_lt8.sample(n=num_examples, random_state=42)
             ))
```

```
=== 20 ===
The classification rule seems to be: statements which do not contain a question or a target
ed demand/address towards someone are labeled as 'True', while those containing a question
or a direct request/order toward someone are labeled as 'False'.
=== 35 ===
The classification rule could be that sentences with terms of address such as 'Master' or s
entences that depict a direct speech or conversation are labeled as 'True'. Statements that
depict general observations, requests, reflections or are part of a narrative are labeled a
s 'False'.
=== 40 ===
The classification rule seems to be that sentences with more complex concepts or propositio
ns are labelled as false, while simpler and straightforward sentences are labelled as true.
The 'true' statements are also more direct, while the 'false' ones involve more dependent c
lauses and indirect propositions.
=== 80 ===
The classification rule appears to be based on whether the statement is a question or not.
Statements labelled as "True" are not in the form of a question while statements labelled a
s "False" are typically questions or sentences with a more complex structure.
=== 100 ===
The classification rule is focused on the grammatical correctness and coherence of the stat
ement. Statements that are grammatically correct and make logical sense are labelled as "Tr
ue", while statements that are grammatically incorrect, incomplete, or do not make sense ar
e labelled as "False".
```

Figure 8: Classification rules articulated by GPT-4 for a dataset labelled using word-counting. Note that for < 5 words at 80 examples, the LLM stumbles on the correct rule.

Again, hinting (with "consider the number of words") is enough to trigger GPT-4 into giving more correct rules, although the precise value of the threshold given varies slightly (e.g. $3-4$ for <5, $6-7$ for <8), possibly due to limited number of examples.

For odd vs even, classification and articulation are both unsuccessful without hinting. See the Jupyter notebook for more details.

4 Bible version classification (inc. Spanish)

We now look at three different translations of the Bible:

- King James Version (KJV): One of the most famous and historically significant English translations of the Bible, originally published in 1611.
- Easy-to-Read Version (ERV): Designed to be easy to read and understand, it's often used for English language learners, children, and people with lower reading skills.
- Reina-Valera 1989 (RV): A modern revision of the classic Spanish Protestant translation that aims to preserve the traditional linguistic beauty while updating the language for contemporary readers.

Since they are all translations of the same text, the classification datasets are balanced.

Classification results are shown in Figure 9, and articulated rules for KJV vs ERV are shown in Figure 10.

- KJV vs RV: Classification is very quickly reacked at 100% accuracy, and articulated rules quickly identify "Spanish vs English".
- KJV vs ERV: Classification is poor, yet GPT-4 manages to correctly articulate the rule "direct quote from the Bible [KJV] vs paraphrase".

It's surprising to me that it can articulate this rule, but can't seem to use it to get good classification. This seems to go in the other direction to the lower-case function, where classification without articulation performed better.

If I were to spend more time on this, I'd investigate GPT-4's ability to identify direct quotes from the KJV when directly prompted, to determine to what extent the poor classification performance comes simply from not having memorised the full KJV.

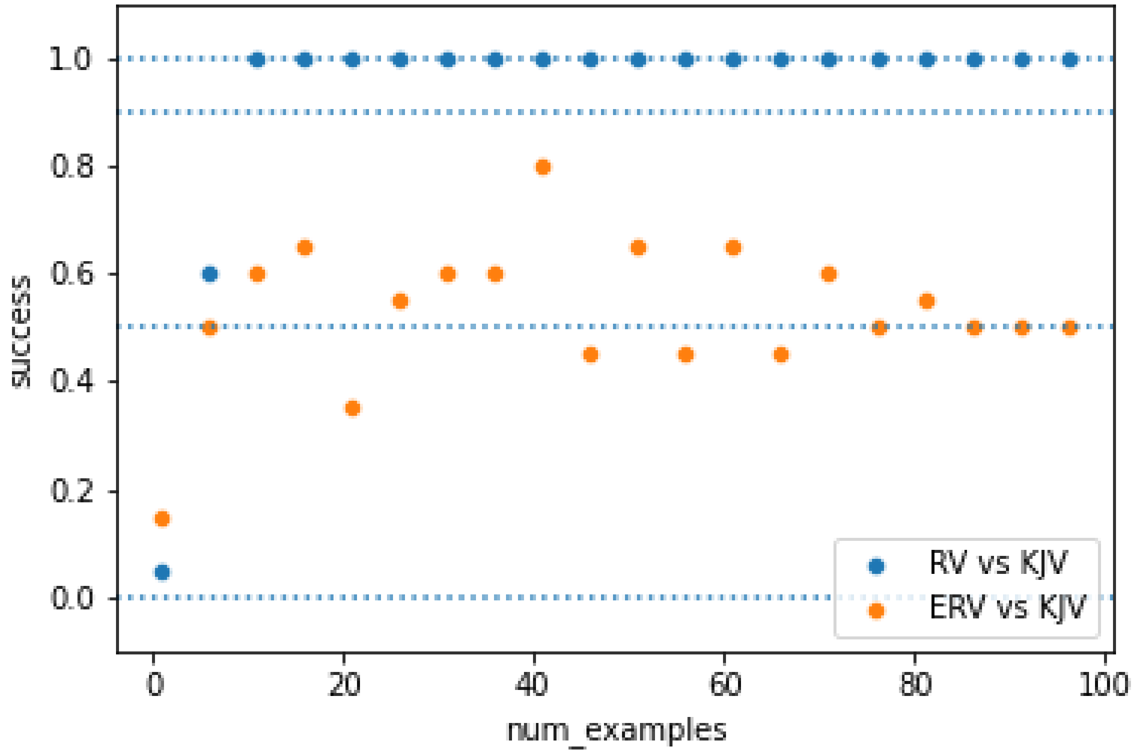
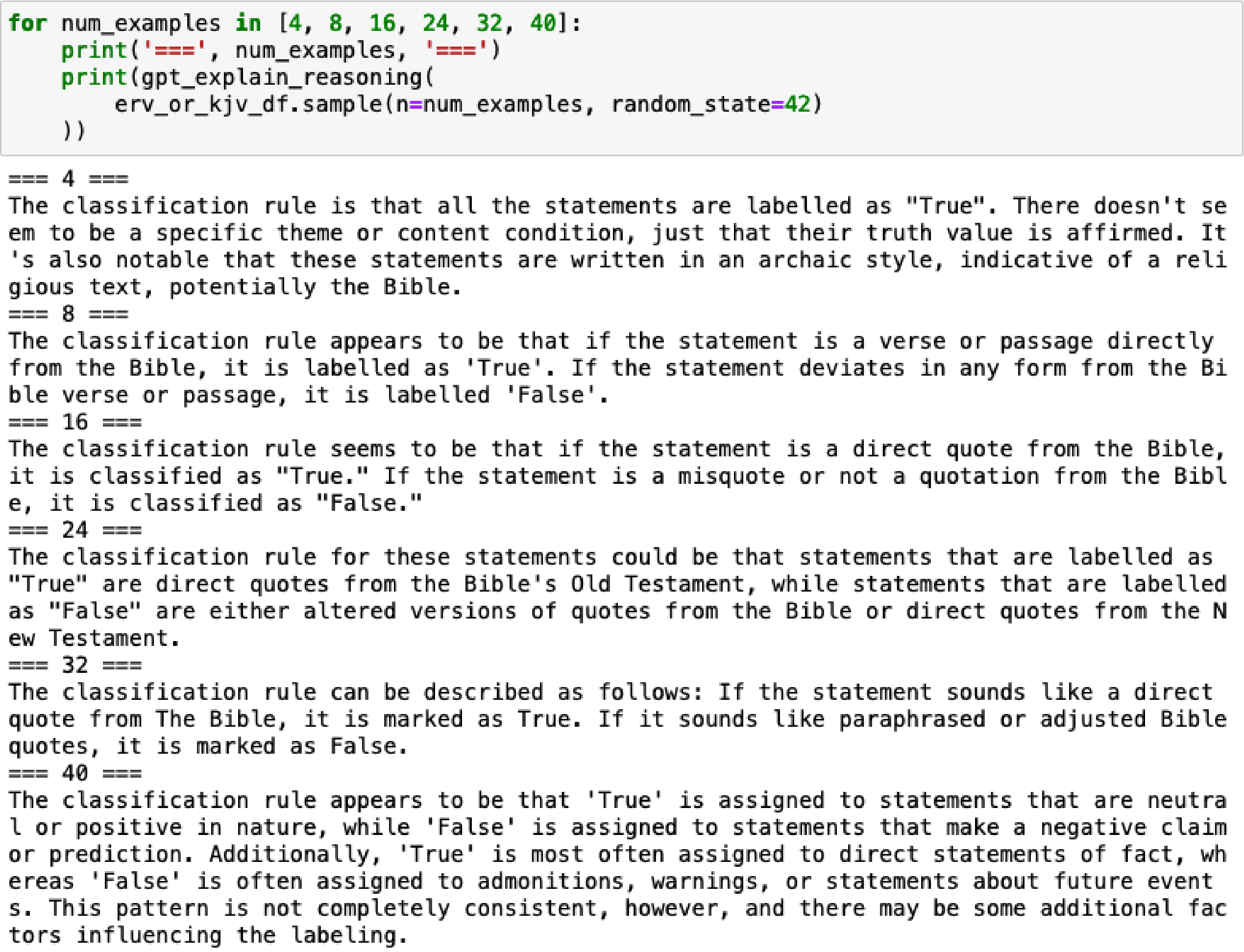


Figure 9: GPT-4 very quickly learns to distinguish verses from a Spanish translation of the Bible (Reina-Valera 1989) from those of an English translation (King James Version), but has trouble distinguishing between the King James Version and another English-language translation in a different style (Easy Reading Version).



```
for num_examples in [4, 8, 16, 24, 32, 40]:
    print('===', num_examples, '===')
    print(gpt_explain_reasoning(
        erv_or_kjv_df.sample(n=num_examples, random_state=42)
    ))
```

```
=== 4 ===
The classification rule is that all the statements are labelled as "True". There doesn't se
em to be a specific theme or content condition, just that their truth value is affirmed. It
's also notable that these statements are written in an archaic style, indicative of a reli
gious text, potentially the Bible.
=== 8 ===
The classification rule appears to be that if the statement is a verse or passage directly
from the Bible, it is labelled as 'True'. If the statement deviates in any form from the Bi
ble verse or passage, it is labelled 'False'.
=== 16 ===
The classification rule seems to be that if the statement is a direct quote from the Bible,
it is classified as "True." If the statement is a misquote or not a quotation from the Bibl
e, it is classified as "False."
=== 24 ===
The classification rule for these statements could be that statements that are labelled as
"True" are direct quotes from the Bible's Old Testament, while statements that are labelled
as "False" are either altered versions of quotes from the Bible or direct quotes from the N
ew Testament.
=== 32 ===
The classification rule can be described as follows: If the statement sounds like a direct
quote from The Bible, it is marked as True. If it sounds like paraphrased or adjusted Bible
quotes, it is marked as False.
=== 40 ===
The classification rule appears to be that 'True' is assigned to statements that are neutra
l or positive in nature, while 'False' is assigned to statements that make a negative claim
or prediction. Additionally, 'True' is most often assigned to direct statements of fact, wh
ereas 'False' is often assigned to admonitions, warnings, or statements about future event
s. This pattern is not completely consistent, however, and there may be some additional fac
tors influencing the labeling.
```

Figure 10: GPT-4 quickly identifies that one class corresponds to “direct quotes” from the bible (presumably King James Version), but describes “paraphrasing” rather than identifying the Easy Reading Version.