Designing Informative Metrics for Few-Shot Example Selection

Anonymous ACL submission

Abstract

Pretrained language models (PLMs) have shown remarkable few-shot learning capabilities when provided with properly formatted examples. However selecting the best examples remains an open challenge. We propose a complexity based prompt selection approach for sequence tagging tasks. This approach avoids the training of a model for selection of examples and instead uses certain metrics to align the syntactico-semantic complexity of test sentences and examples. We use both sentence- and word-level metrics to match the complexity of examples to the (test) sentence being considered. Our results demonstrate that our approach extracts greater capability from PLMs: it achieves state-of-the-art performance on few-shot NER, achieving a 5% absolute improvement in F1 score on the CoNLL2003 dataset for `GPT-4`. We also see large gains of upto 28.85 points (F1/Acc.) in smaller models like `GPT-j-6B`.[1]

1 Introduction

Pretrained language models (PLMs) have demonstrated impressive few-shot learning capabilities when prompted with examples for a particular task (Brown et al., 2020). In few-shot prompting, k examples that demonstrate the goal of the downstream task along with the test sample (collectively called the k-shot prompt) are provided to the model. But the effectiveness of PLMs in few-shot learning depends on the quality of these prompts; this is challenging due to the need for unambiguous instruction (Sclar et al., 2024) which is further complicated due to the task of selecting examples. Appropriately selected examples will ensure that the model can generalize effectively and perform optimally across diverse natural language processing (NLP) tasks.

Recent work has shown that specific examples provided in-context can significantly impact the performance of PLMs (Min et al., 2022). Sorensen et al. (2022) notes that specific example combinations lead to better generalization, and connects this with the informativeness of these examples. This suggests that there are certain metrics that can be used to capture properties of prompts that result in improved performance. Most prior work for prompt selection has either trained a prompt retriever to select examples, or used PLMs themselves to label examples (Rubin et al., 2022) to serve as a good prompt. However, both approaches require extra compute and task specific training. Further, Liu et al. (2022) uses nearest neighbours algorithms to select examples. However, this solely relies on embedding space properties, and not consider other metrics related to the label distribution.

In this work, we propose a new approach for example selection for sequence tagging tasks. These tasks are important because they enable the extraction of structured information from unstructured text data; this serves as a fundamental building block for various NLP tasks. Our approach, *complexity based prompt retrieval (CS retrieval)*, aims to select examples from the training data whose "properties" align with that of the test sentence. We use sentence- and word-level complexity measures to match prompt examples to evaluation sentences.

We conduct our experiments on Named entity recognition (NER), Part-of-speech (POS) tagging, Sentence Chunking. We also devise a new dataset for privacy policies involving tagging of Contextual Integrity parameters (Shvartzshnaider et al., 2019). We show that CS retrieval improves accuracy significantly in all cases.

The contributions of this work are as follows:

1. We propose CS retrieval to select prompts based on sentence and word level complexity metrics (§ 3).

[1] Structured prompting base code taken from https://github.com/blvns/structured_prompting.git . We will release our code upon acceptance of the paper.

2. We demonstrate improved few-shot accuracy with CS retrieval. Our approach achieves state-of-the-art performance in few-shot NER with a 5% absolute improvement in F1 score on the CoNLL2003 dataset for GPT-4 and upto 28.85 points (F1/Acc.) in smaller models (§ 5).

2 Related Work

To perform our experiments, we use the *Structured Prompting* paradigm introduced by Blevins et al. (2023) for named entity recognition (NER), part-of-speech (POS) tagging, and sentence chunking. Each model that we use is provided k pairs of (sentence, tagged sentence) as the task examples in a k-shot prompt. This involves iteratively feeding the PLM a word and its predicted label to get it to label the next word. These are sequence tagging tasks: those that assign labels or tags to tokens or other units of text. Experiments show that PLMs can effectively solve these tasks with just a few demonstrations using this technique. We use this method as our baseline.

Liu et al. (2022) has explored strategies for selecting examples to leverage GPT-3's few-shot learning capabilities. They analyzed the sensitivity of GPT-3's performance to randomly sampled examples and found significant variability which is also detailed in Min et al. (2022). They proposed a retrieval based approach called KATE that selects semantically similar examples by using k-nearest neighbors in the embedding space of pre-trained sentence encoders like BERT and RoBERTa. They found that KATE substantially improved GPT-3's results over random sampling on natural language understanding and generation tasks.

3 Methodology

Our core idea is simple: to select examples that closely resemble the test sample in both semantics and length while ensuring a diverse range of labels for the task. We measure the "complexity" of a candidate example using 3 different sentence-level and word-level metrics: (a) normalized sentence similarity score, (b) normalized smoothed length similarity, and (c) normalized label entropy.

Problem Setup: Assume we have a training dataset of the form $D = \{z_i = (x_i, y_i)\}_{i=1}^{n}$ and a test example x_{test}, our goal is to use a complexity score to select a subset of examples S of the training dataset ($S \subset D$) s.t. $|S| = k$.

3.1 Normalized Sentence Similarity Score

For a given test sentence, we calculate its similarity scores with each candidate sentence for the k-shot prompt. We convert each of the sentences into sentence embeddings using the all-MiniLM-L6-v2 sentence transformer (results do not vary drastically with alternate transformers). This produces an embedding size of 384 dimensions. To find the similarity score, we calculate cosine similarity between the embeddings.

$$\texttt{Similarity}(x_i, x_{test}) = \texttt{cosine_sim}(\texttt{emb}(x_i), \texttt{emb}(x_{test}))$$

These values are normalized to get our final sentence similarity scores. Formally, this is denoted as NormSentSimilarity[2].

3.2 Normalized Smoothed Length Similarity

This metric is based on the difference in sentence lengths between the candidate sentence and the test sentence. We use this metric since cosine similarity of sentence embeddings does not inherently factor sentence length. The smoothed length similarity (SLS) transforms the absolute length difference between two sentences using a sigmoid function to produce a smooth, tapered similarity curve rather than a hard threshold. The sigmoidal nature of SLS prevents stark drops in scores for small length divergences. This is important as, for few-shot prompting, some flexibility in length ranges is desirable to provide syntactic variety while maintaining coarse length matching. It is formulated as follows:

$$\texttt{SLS}(x_i, x_{test}) = (1 + \exp(\frac{|\texttt{len}(x_i) - \texttt{len}(x_{test})|}{T}))^{-1}$$

where T controls the sigmoidal shape (T = 3 in our experiments). Sentences with similar lengths achieve higher SLS scores, while increasingly divergent lengths taper down towards 0 in a continuous, non-binary fashion. These values are then normalized to provide our final Normalized Smoothed Length Similarity scores denoted as NormSLS[2].

3.3 Normalized Label Entropy

For a given task, let $\mathcal{Y}$ define the space of possible labels. For sentence x_i, let y_i be the list of labels, and $\nu(y_i^j)$ is the frequency of label $j \in \mathcal{Y}$ in list y_i. We can calculate $\Pr(y_i^j) = \nu(y_i^j)/|x_i|$. Entropy is then calculated as:

$$\mathrm{H}(y_i) = -\sum_{j \in \mathcal{Y}} \Pr(y_i^j) \log_2 \Pr(y_i^j)$$

[2]Normalization here is N(s_i) = s_i / max(S) where $S = \{s_1, \cdots, s_n\}$ and $s_i \in \mathbb{R}$

The idea is that sentences where labels are more skewed provide less information gain for in-context learning compared to flatter distributions. Entropy can be used to quantify that skew difference. We normalize these entropy values to get our final NormEntropy[2] scores.

3.4 Complexity Score

We propose a complexity score to align the syntactico-semantic complexity of prompts and examples. The three component metrics are weighted and summed to produce the final complexity score:

$$\begin{aligned}\texttt{CS}(z_i, x_{test}) &= w_1 * \texttt{NormSLS}(x_i, x_{test}) \\ &+ w_2 * \texttt{NormEntropy}(y_i) \\ &+ w_3 * \texttt{NormSentSimilarity}(x_i, x_{test})\end{aligned}$$

We use this score to select the k highest scoring train sentences for a k-shot prompt. w_1, w_2, w_3 are weights that are set using grid search to optimize scores on the development set for each sequence tagging task. For example, we found that $(w_1, w_2, w_3) = (0.25, 0.25, 0.5)$ is the best value for NER.

4 Experimental Setup

4.1 Tasks

We perform our experiments on 4 different sequence tagging tasks namely NER, POS tagging, sentence chunking and on a custom dataset of privacy policies for tagging contextual integrity (CI) parameters (Shvartzshanider et al., 2023). Each task involves the annotation of tokens in a sequence with specific labels, facilitating the extraction of valuable information and enhancing language understanding. For all datasets, we use a random set of 1000 test samples. For the CI task, we use 89 samples.

1. NER: This focuses on classifying named entities within a text. The entities can range from persons and organization to locations and dates. For our experiments, we use the widely recognized CoNLL2003 dataset (Tjong Kim Sang and De Meulder, 2003) for measuring the proficiency of our CS retrieval.

2. POS Tagging: This involves assigning grammatical categories (e.g., nouns, verbs, adjectives) to each word in a sentence. We evaluate the POS tagging performance using the English Universal Dependencies (UD) treebank annotated on the GUM corpus (Zeldes, 2017). It employs the UPOS tagset introduced by Nivre et al. (2020).

3. Sentence Chunking: This aims to partition sentences into non-overlapping syntactic units. For our experiments, we use the CONLL2000 dataset (Tjong Kim Sang and Buchholz, 2000) to frame chunking as a BIO tagging task.

4. CI parameters: We additionally created a custom dataset (which we will release) that involves the sequence tagging task of CI parameters for privacy policies which is taken from Shvartzshanider et al. (2023). CI parameters sequence tagging is a crucial task within the realm of privacy-aware NLP (Shvartzshnaider et al., 2019).

4.2 Models

We perform our experiments on the GPT-Neo series of models which has parameter counts from 125 million to 20 billion (GPT-Neo-125M, GPT-Neo-1.3B, GPT-Neo-2.7B, GPT-j-6B, GPT-NeoX-20B) and additionally 2 black box models (Davinci-002 and GPT-4) for the sequence tagging tasks of NER, POS Tagging and Sentence Chunking. For our custom dataset for CI parameters tagging, we use Llama2 variants.

5 Results and Discussions

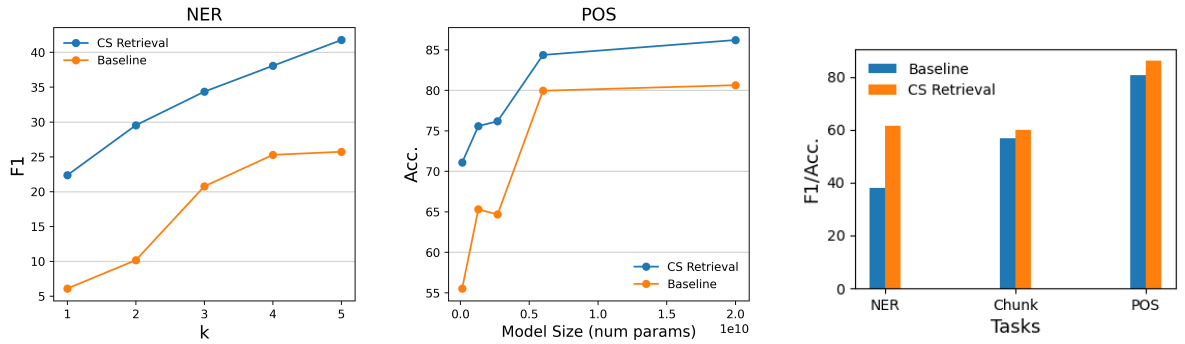
Our results show that CS retrieval significantly improves few-shot accuracy over the baseline across all models and considered tasks. Table 1 compares the F1 and Accuracy scores (for NER, Chunking and POS) for different models while keeping the number of examples constant. Table 2 provides the same for the CI tagging task on Llama models.

We observe substantial gains using this method with the largest gains on GPT-j-6B which achieves a 28.85 point increase on NER, GPT-Neo-2.7B which achieves a 24.82 point increase in chunking and GPT-Neo-125M which achieves a 15.56 point increase in POS tagging. It is also evident that the gains of example selection drops with increasing model size in figures 1(b) and 1(c) (with GPT-4 having the least gains) and this is due to the method approaching overall accuracy saturation points. This is also combined with the fact that larger models have seen more diverse training data, so random prompt selections are more than likely to cover the needed distribution and hence, additional gains from optimal selection are diminished. However, even with small gains, we achieve state-of-the-art

[3] Accuracy has changed with respect to the original paper for this model probably because of the the reasoning in Chen et al. (2023)

	NER (F1)		Chunking (F1)		POS (Acc.)	
	Baseline	CS retrieval	Baseline	CS retrieval	Baseline	CS retrieval
GPT-Neo-125M	12.46	**29.21**	24.28	**40.33**	55.53	**71.09**
GPT-Neo-1.3B	31.49	**52.44**	25.44	**44.19**	65.30	**75.58**
GPT-Neo-2.7B	25.73	**41.77**	28.50	**53.32**	64.68	**76.19**
GPT-j-6B	25.88	**54.73**	35.85	**54.56**	79.96	**84.37**
GPT-Neox-20B	37.74	**61.49**	56.18	**59.66**	80.64	**86.21**
Davinci-002 (Black Box)	20.30	**40.03**	35.82	**51.79**	46.04[3]	**52.20**
GPT-4 (Black Box)	83.48*	**88.76**	79.78	**84.52**	92.93	**93.47**

Table 1: Results of using Complexity Score for retrieval (CS retrieval) of the best scoring k examples for the few-shot prompt (here $k = 5$). The baseline is in the structured prompting paradigm (Blevins et al., 2023) with static prompt examples. CS retrieval significantly surpasses the baseline using linguistic structure alone for the prompt (details in Appendix A.1) in all cases. Best values are in boldface. * in the table shows the previously achieved state-of-the-art performance on few-shot NER by Ashok and Lipton (2023).



(a) (b) (c)

Figure 1: We evaluate NER and Chunking on F1 and POS on Accuracy. (a) Performance while varying number of demonstrations k (GPT-Neo-2.7B) for NER. (b) Performance while varying model size for a fixed $k = 5$ for POS. (c) Performance for the various tasks on GPT-NeoX-20B model with $k = 5$.

	CI tagging (Acc.)	
	Baseline	CS retrieval
Llama2-13B	37.56	**56.79**
Llama2-70B	43.90	**59.35**

Table 2: Results of using complexity score for retrieval (CS retrieval) on the CI parameters tagging task using Llama2 models. Best values are in boldface.

performance on the CoNLL2003 dataset with a 5% increase over PromptNER methodology (Ashok and Lipton, 2023) for few-shot NER .

From Figure 1(a), we see that our approach enables effective learning (larger gains) with smaller k. At $k = 2$, we see a gain of 19.37 points over the baseline which decreased when k is increased. The percentage gains consistently reduce as we increase the number of examples.

While we have focused on complexity metrics for selecting examples in simple prompting setups, what we want to put forth is that this approach can further boost performance when combined with more sophisticated prompting techniques. For example, CS retrieval can be used along with chain-of-thought (CoT) prompting (Wei et al., 2023). It can be used to identify the most information-rich examples for the initial context in CoT prompting which will result in larger gains.

6 Conclusion

In summary, CS retrieval is a flexible technique that can be used to enhance the accuracy of existing few-shot approaches using example scoring. Additionally, it provides a task-agnostic quantification of the informativeness of examples for improved model performance. The relative weights of the components of the complexity score can be tuned on a per-task basis. Our approach allows us to extract the most out PLMs without any finetuning of the parameters of the model.

7 Limitations

Our methodology focuses on sequence tagging tasks and cannot be applied to tasks outside this domain such as question answering. This is because the normalized entropy metric is specifically designed for encouraging selection of sentences with a wider label variety on the token level; this helps increase the sampling of underrepresented labels. Further, we have conducted all experiments for the sequence tagging tasks in only English language. Performance with other languages has not been analyzed.

References

Dhananjay Ashok and Zachary C. Lipton. 2023. Promptner: Prompting for named entity recognition.

Terra Blevins, Hila Gonen, and Luke Zettlemoyer. 2023. Prompting language models for linguistic structure.

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.

Lingjiao Chen, Matei Zaharia, and James Zou. 2023. How is chatgpt's behavior changing over time?

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work?

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. In *The Twelfth International Conference on Learning Representations*.

Yan Shvartzshanider, Ananth Balashankar, Thomas Wies, and Lakshminarayanan Subramanian. 2023. Beyond the text: Analysis of privacy statements through syntactic and semantic role labeling. In *Proceedings of the Natural Legal Language Processing Workshop 2023*, pages 85–98, Singapore. Association for Computational Linguistics.

Yan Shvartzshnaider, Noah J. Apthorpe, Nick Feamster, and Helen Nissenbaum. 2019. Going against the (appropriate) flow: A contextual integrity approach to privacy policy analysis. In *AAAI Conference on Human Computation & Crowdsourcing*.

Taylor Sorensen, Joshua Robinson, Christopher Rytting, Alexander Shaw, Kyle Rogers, Alexia Delorey, Mahmoud Khalil, Nancy Fulda, and David Wingate. 2022. An information-theoretic approach to prompt engineering without ground truth labels. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 shared task chunking. In *Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Amir Zeldes. 2017. The gum corpus: creating multilayer resources in the classroom. *Lang. Resour. Eval.*, 51(3):581–612.

A Appendix

A.1 Example Prompts

Context: The Finance Ministry raised the price for tap sales of the Dutch government 's new 5.75 percent bond due September 2002 to 99.95 from 99.90 .
Tagged: The_O Finance_B-ORG Ministry_I-ORG raised_O the_O price_O for_O tap_O sales_O of_O the_O Dutch_B-MISC government_O 's_O new_O 5.75_O percent_O bond_O due_O September_O 2002_O to_O 99.95_O from_O 99.90_O ._O
Context: Swiss bonds ended mostly higher in generally quiet activity , with the September confederate bond futures contract holding just above 113.00 .
Tagged: Swiss_B-MISC bonds_O ended_O mostly_O higher_O in_O generally_O quiet_O activity_O ,_O with_O the_O September_O confederate_O bond_O futures_O contract_O holding_O just_O above_O 113.00_O ._O
Context: The Brent crude futures market on the Singapore International Monetary Exchange (SIMEX) was closed on Monday in respect for a U.K. national holiday .
Tagged: The_O Brent_B-ORG crude_O futures_O market_O on_O the_O Singapore_B-ORG International_I-ORG Monetary_I-ORG Exchange_I-ORG (_O SIMEX_B-ORG)_O was_O closed_O on_O Monday_O in_O respect_O for_O a_O U.K._B-LOC national_O holiday_O ._O
Context: European bourses closed mixed on Tuesday with London clawing back most of the day 's losses despite an unsteady start on wall Street , hit by inflation worries .
Tagged: European_B-MISC bourses_O closed_O mixed_O on_O Tuesday_O with_O London_B-LOC clawing_O back_O most_O of_O the_O day_O 's_O losses_O despite_O an_O unsteady_O start_O on_O wall_B-ORG Street_I-ORG ,_O hit_O by_O inflation_O worries_O ._O
Context: No closures of airports in the Commonwealth of Independent States are expected on August 24 and August 25 , the Russian Weather Service said on Friday .
Tagged: No_O closures_O of_O airports_O in_O the_O Commonwealth_B-LOC of_I-LOC Independent_I-LOC States_I-LOC are_O expected_O on_O August_O 24_O and_O August_O 25_O ,_O the_O Russian_B-ORG Weather_I-ORG Service_I-ORG said_O on_O Friday_O ._O
Context: Hungarian overnight interest rates closed higher on Friday as market liquidity tightened before the December 10 social security contribution payment deadline, dealers said.
Tagged:

Figure 2: Prompt generated for a specific test sample after retrieval of 5 examples (5-shot prompt) from the training dataset for **Named Entity Recognition** using our approach. The test sentence that we want the model to label is provided at the end of the prompt and is shown in bold

Context: Shearson is offering the notes as 6 34 % securities priced to yield 6.15 % .
Tagged: Shearson_B-NP is_B-VP offering_I-VP the_B-NP notes_I-NP as_B-PP 6_B-NP 34_I-NP %_I-NP securities_I-NP priced_B-VP to_B-VP yield_I-VP 6.15_B-NP %_I-NP ._O
Context: Bonds due 1991-1996 carry 6.70 % coupons and bonds due 1997-2000 carry 6 34 % coupons .
Tagged: Bonds_B-NP due_B-ADJP 1991-1996_B-NP carry_B-VP 6.70_B-NP %_I-NP coupons_I-NP and_O bonds_B-NP due_B-ADJP 1997-2000_B-NP carry_B-VP 6_B-NP 34_I-NP %_I-NP coupons_I-NP ._O
Context: Their price falls further than that of other bonds when inflation and interest rates kick up .
Tagged: Their_B-NP price_I-NP falls_B-VP further_B-ADVP than_B-PP that_B-NP of_B-PP other_B-NP bonds_I-NP when_B-ADVP inflation_B-NP and_O interest_B-NP rates_I-NP kick_B-VP up_B-PRT ._O
Context: Serial bonds were priced at par to yield to 6.90 % in 2000 .
Tagged: Serial_B-NP bonds_I-NP were_B-VP priced_I-VP at_B-PP par_B-NP to_B-VP yield_I-VP to_B-PP 6.90_B-NP %_I-NP in_B-PP 2000_B-NP ._O
Context: At the auction of six-month U.S. Treasury bills on Monday , the average yield fell to 7.61 % from 7.82 % .
Tagged: At_B-PP the_B-NP auction_I-NP of_B-PP six-month_B-NP U.S._I-NP Treasury_I-NP bills_I-NP on_B-PP Monday_B-NP ,_O the_B-NP average_I-NP yield_I-NP fell_B-VP to_B-PP 7.61_B-NP %_I-NP from_B-PP 7.82_B-NP %_I-NP ._O
Context: The rate on six-month bills rose to 7.53 % for a bond-equivalent yield of 7.92 % .
Tagged:

Figure 3: Prompt generated for a specific test sample after retrieval of 5 examples (5-shot prompt) from the training dataset for **Sentence Chunking** using our approach. The test sentence that we want the model to label is provided at the end of the prompt and is shown in bold

Context: Many forms of culture are passed down through a combination of deliberate and unconscious processes .
Tagged: Many_ADJ forms_NOUN of_ADP culture_NOUN are_AUX passed_VERB down_ADP through_ADP a_DET combination_NOUN of_ADP deliberate_ADJ and_CCONJ unconscious_ADJ processes_NOUN ._PUNCT
Context: Cognitive psychology is the field of psychology dedicated to examining how people think .
Tagged: Cognitive_ADJ psychology_NOUN is_AUX the_DET field_NOUN of_ADP psychology_NOUN dedicated_VERB to_SCONJ examining_VERB how_ADV people_NOUN think_VERB ._PUNCT
Context: Having been a psychologist for a number of years gives me a leg up on it .
Tagged: Having_AUX been_AUX a_DET psychologist_NOUN for_ADP a_DET number_NOUN of_ADP years_NOUN gives_VERB me_PRON a_DET leg_NOUN up_ADV on_ADP it_PRON ._PUNCT
Context: Do they get mad or irritated if the centre of attention moves to someone else ?
Tagged: Do_AUX they_PRON get_VERB mad_ADJ or_CCONJ irritated_VERB if_SCONJ the_DET centre_NOUN of_ADP attention_NOUN moves_VERB to_ADP someone_PRON else_ADV ?_PUNCT
Context: When thoughts are formed , the brain also pulls information from emotions and memories (Figure 7.2) .
Tagged: When_ADV thoughts_NOUN are_AUX formed_VERB ,_PUNCT the_DET brain_NOUN also_ADV pulls_VERB information_NOUN from_ADP emotions_NOUN and_CCONJ memories_NOUN (_PUNCT Figure_PROPN 7.2_NUM)_PUNCT ._PUNCT
Context: Psychologists have examined the mental processes that underpin conscious and unconscious biases [6] ;
Tagged:

Figure 4: Prompt generated for a specific test sample after retrieval of 5 examples (5-shot prompt) from the training dataset for **Part-of-Speech** tagging using our approach. The test sentence that we want the model to label is provided at the end of the prompt and is shown in bold