Semantics or Syntax? Which is More Important in In-Context Learning for Sentence Classification

Anonymous ACL submission

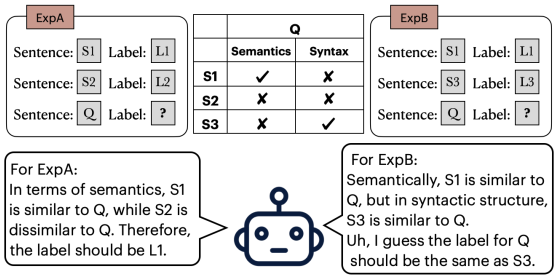
Abstract

In this study, we explore the impact of semantics and syntax in the construction of demonstration examples for in-context learning (ICL) with Large Language Models (LLMs). We identify the limitations of current methods that prioritize semantic similarity and underscore the importance of syntactic information, which has been underrepresented in sentence-level classification tasks. Through experiments measuring semantic and syntactic similarities, we reveal that ICL methods tend to favor syntactic congruence. Consequently, we propose a novel Semantics and Syntax-based Sentence Selection (SSSS) framework for selecting demonstration examples in ICL, integrating both semantic and syntactic dimensions. This approach addresses the challenges of constructing accurate semantic representations and quantifying syntactic structure similarities. The experimental results on three datasets suggest that the SSSS approach can facilitate more effective ICL by incorporating syntax into the demonstrative example selection, potentially leading to enhanced model performance.

1 Introduction

Large Language Models (LLMs) exhibit robust performance across a broad spectrum of tasks (Brown et al., 2020; Chowdhery et al., 2023). In-context learning (ICL) has revealed the proficiency of LLMs in task adaptation through minimal example-driven demonstrative contexts. Arranged as input-label pairings within natural language templates, these exemplars form the foundation of ICL. Owing to its straightforwardness and efficacy, ICL is extensively employed in a multitude of Natural Language Processing (NLP) tasks.

However, the ICL is significant sensitivity to the choice of demonstration examples induces notable performance variations (Zhao et al., 2021). Previous research efforts have concentrated on curating optimal context exemplars for test cases via semantic distributions (Dong et al., 2023). This typically involves employing sentence semantics encoder models to ascertain test sentence similarity with a training set, thereby selecting the comparably most resembling examples. While improvements are evident when utilizing semantic embedding-based selection methods, they may fall short for tasks where syntactic structure is paramount. Tasks such as Relation Classification (Liu et al., 2015) and Question Classification (Ma et al., 2015) have demonstrated how syntactic cues substantially contribute to sentence analysis. Despite their potential, these syntactic details are frequently overlooked in the era of LLMs' ICL application. Addressing this gap, our research asks whether the construction of demonstrations is more influenced by semantic or syntactic elements, or if an integrative approach is preferable. Our preliminary experiments, including semantic and syntactic similarity assessments, indicate a predilection of ICL methods towards syntactic parallels. We also propose an information flow method at the sample level to examine attention shifts with varying demonstrations, reinforcing our syntactic emphasis hypothesis. These findings



	Q	
	Semantics	Syntax
S1	✓	✗
S2	✗	✗
S3	✗	✓

For ExpA:
In terms of semantics, S1 is similar to Q, while S2 is dissimilar to Q. Therefore, the label should be L1.

For ExpB:
Semantically, S1 is similar to Q, but in syntactic structure, S3 is similar to Q.
Uh, I guess the label for Q should be the same as S3.

Figure 1: A binary classification task with one positive and one negative example as the demonstration. We adopt two strategies to select examples with different semantic and syntactic similarities. The experimental results show that ICL methods tend to favor sentences with higher syntactic similarity.

advocate for the integration of syntactic attributes in creating demonstrative examples to refine ICL performance.

In light of these insights, we introduce a Semantics and Syntax-based Sentence Selection (SSSS) approach for ICL with LLMs. The SSSS approach independently assesses semantic and syntactic similarities before combining them to guide example selection. The proposed approach addresses two challenges: (1) From the semantic perspective – How to construct semantic representations for sentences? (2) From the syntactic perspective – How to accurately describe the syntactic structure of sentences and measure the syntactic structure similarity between two sentences? For the first challenge, we focus on developing a semantic representation model using relational data. For the latter, a dependency parsing model produces parsing trees for sentences, with syntactic tree similarities determined through the tree edit distance algorithm.

Our contributions are summarized as follows:

1. We identify and empirically verify a discernable preference by Large Language Models (LLMs) within in-context learning (ICL) methods for demonstration sentences mirroring the syntactic structure of the test input. This syntactic alignment preference is further substantiated through an information flow analysis predicated on attention weights.

2. We propose a hybridized approach, **S**emantics and **S**yntax based **S**entence **S**election(SSSS), which concurrently leverages semantic content and syntactic structuring to formulate demonstrative examples with greater efficacy.

3. To assess the practical utility of the SSSS approach, we conduct experiments on three datasets among two tasks. Experimental results demonstrate the superior performance of our approach in enhancing ICL capabilities.

2 Preliminary Exploration

2.1 A Binary Classification Task

To explore whether semantic or syntactic features are more important in construction the demonstrations for ICL method using LLMs, we initially construct a binary classification task. Specifically, we first select the two relation categories with the highest number of sentences in the training set of Re-TACRED (per:identity and per:title). And then we randomly sample 100 sentences for both relations as the test set, while the remaining data served as the training data pool. In order to compare the performance fairly, we totally select 2k (k=2,4,8,16) examples for each test sentence in which k examples are positive while another k examples are negative (depends on whether the label is same or different with the label of test sentence). Therefore, the positive examples are semantic similar while negative examples are semantic dissimilar. As for syntactic features, we utilize a Tree Edit Distance (TED) algorithm to evaluate the syntactic similarity on the dependency parsing trees of sentences. This content is detailed in Sec 3.2. Finally, as shown in Tab 1, we obtain four different sets of demonstrations for each sentence to test the performance of ICL.

Method	Pos		Neg	
	Sem	Syn	Sem	Syn
RanPos+RanNeg	✓	✗	✗	✗
RanPos+SynNeg	✓	✗	✗	✓
SynPos+RanNeg	✓	✓	✗	✗
SynPos+SynNeg	✓	✓	✗	✓

Table 1: Data selection strategies for four comparative experiments, where ‘Sem’ and ‘Syn’ respectively indicate whether the examples are similar in terms of semantics and syntax.

2.2 Semantic or Syntactic?

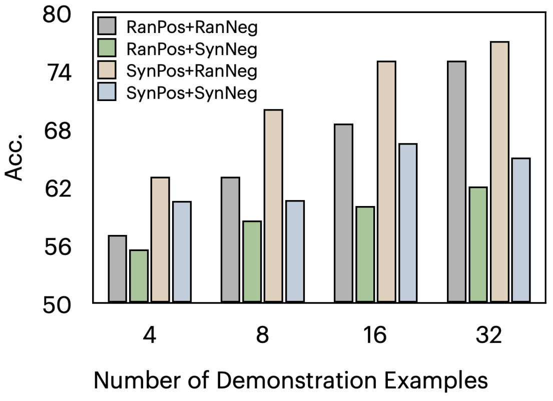


Figure 2: Results of Binary Task.

For the binary classification task, we test the accuracy of the ICL prediction on LLaMa2-7b-chat model under four different demonstration selection methods. The experimental results are shown in Figure 2, where the horizontal axis represents the number of context examples, the vertical axis represents the accuracy.

RanPos+RanNeg vs RanPos+SynNeg: When there is only a semantic difference, that is, with RanPos+RanNeg, the performance of the binary classification task exceeds 50%. Moreover, as the number of context examples increases, the model's performance gradually improves. This indicates that the ICL method can make correct predictions based on semantic similarity. However, there is a noticeable decline when negative examples employ sentences with similar syntax. This result indicates there is a bias for LLMs that towards negative sentences with similar syntactic structures.

RanPos+RanNeg vs SynPos+RanNeg: Conversely, when positive examples incorporate sentences with similar syntax, the model obtains a notable improvement. This suggests that the model tends to lean more towards positive examples when syntactic constraints are added.

RanPos+RanNeg vs SynPos+SynNeg: After adding syntax constraints to both positive and negative examples, it only surpasses the RanPos+RanNeg when the context examples number is 4 but outputs a performance drop on all other settings. This suggests that when both positive and negative examples consist of sentences with similar syntax, it challenging for the model to distinguish semantically similar positive examples from semantically dissimilar negative examples.

Based on the these experiments, we can infer that the ICL method exhibits a bias towards data with similar syntax. However, the introduction of syntax similarity constraints, alongside semantic similarity, leads to a further inclination of the ICL method towards such examples. Consequently, semantic and syntax similarity act in a complementary manner.

2.3 Analysis via Information Flow

To delve deeper into the influence of syntactic similarity features during the ICL process, we draw inspiration from Wang et al. (2023a) and utilize saliency scores to gauge the flow of information within ICL method. As a widely used interpretation tool, the saliency technique (Simonyan et al., 2014) is utilized to emphasize the token interactions. In order to calculate the saliency score for each token of the attention matrix, we following the common practice to employ the Taylor expansion (Michel et al., 2019).

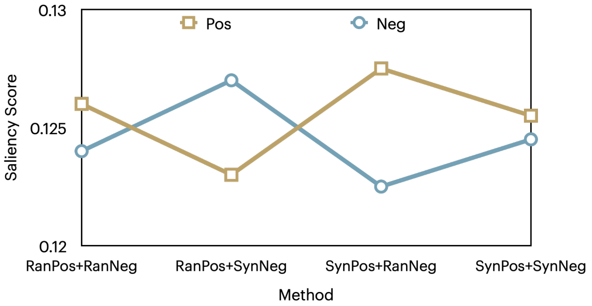


Figure 3: The saliency score of positive and negative examples in demonstration. 'Pos' and 'Neg' represent the normalized averages of all positive and negative example data in the demonstration, respectively.

$$I_l = \sum_h \left| A_{h,l} \odot \frac{\partial \mathcal{L}(x)}{\partial A_{h,l}} \right| \tag{1}$$

Differing from the analysis in Wang et al. (2023a), which examines the separate impacts of words in sentences and labels on the final prediction, we compare the influence of each example in the demonstrations on the final prediction. Thus, we measure the impact of each example on the final token q by averaging influences of all tokens in the example:

$$SaliencyScore_{sq} = \frac{\sum_{(i,j)\in C_{sq}} I_l(i,j)}{|C_{sq}|}, \qquad C_{sq} = \{(q,j) : x_0 \le j \le x_n\} \tag{2}$$

Here, x_0 and x_n represent the positions of the first and last tokens of example s in the demonstration, C_{sq} signifies the influence of each token in s on the final token q, and $SaliencyScore_{sq}$ is the average impact of example s.

In our analysis, we conducte experiments with a demonstration of 8 examples and use the attention weights of the last layer ($l = 32$) in LLaMa2-7b-chat. The saliency scores are averaged for positive and negative sentences, respectively. And they are normalized to assess whether the model tends to positive or negative examples. The analysis results are shown in Figure 3.

From this, we observe that the saliency score variation trend is proportional to the prediction results in Figure 2. Particularly, when using RanPos+SynNeg, the saliency score for negative examples is significantly higher than that for positive examples. Conversely, when using SynPos+RanNeg, the saliency score for positive examples surpasses that for negative examples.

Therefore, this analysis experiment aligns with the above experimental results, consistently indicating that syntactic features play a crucial role in context learning.

3 Methodology

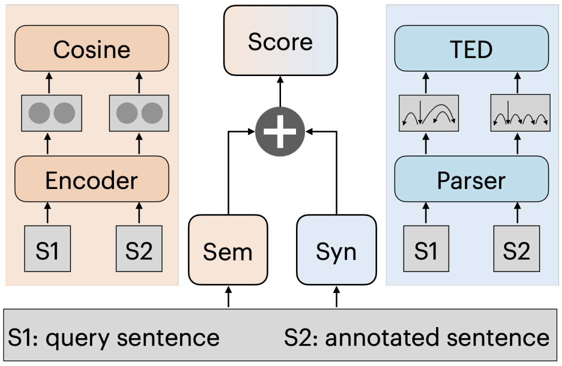


Figure 4: Framework of our SSSS.

According to findings of semantic and syntactic features in the construction of ICL demonstrations, we propose a **S**emantics and **S**yntax based **S**entence **S**election (SSSS) method.

As illustrated in Figure 4, we first obtain the semantic and syntactic representations for sentences, respectively. Then, we calculate both the semantic and syntactic similarities for the sentence pair. Finally, we combine the two similarities to derive a similarity score. For a test sentence, we get the score of each sentence in the training pool and return the top K sentences, which are selected as the context examples for the test sentence.

This section begins by describing the proposed semantic sentence selection method. Then we describe the syntactic representations of sentences and the calculation of syntactic similarity. Finally, we present the method for combing semantic and syntactic similarity.

3.1 Semantics-based Sentence Selection

To obtain the semantic similarity between two sentences, previous works usually employ pre-trained models fine-tuned on Natural Language Inference (NLI) data (Conneau et al., 2017; Reimers and Gurevych, 2019; Gao et al., 2021). In NLI, the entailment relationship assesses whether the meaning of the hypothesis can be logically inferred from the premise.

In this study, we posit that the relationship representation in NLI captures not only semantic similarity but also logical inference. On the other hand, relation extraction data recognizes the semantic similarity between two sentences, regardless of whether one can be logically inferred from the other. Therefore, in this work, we train the sentence semantic representation model using automatically annotated relation extraction datasets.

To train the sentence embedding model, we begin by constructing the training data. Since a sentence can express multiple semantic relations, when building positive and negative examples for the contrastive model, we select positive examples where the sets of candidate relation semantics for two sentences have a non-empty intersection. Conversely, for negative examples, we choose pairs where the sets of candidate relation semantics have no intersection. As defined follows:

$$\begin{aligned} D_{pos} &= \{(s, s^{+}) : |r_s \cap r_{s^{+}}| \geq 1\} \\ D_{neg} &= \{(s, s^{-}) : |r_s \cap r_{s^{-}}| = 1\} \end{aligned} \tag{3}$$

where s^{+} and s^{-} are randomly selected from D for each sentence $s \in D$ until the conditions are met, and r_s represents the set of all possible relation categories for sentence s. For each sentence s, we construct one positive example and one negative example.

Then, we follow the contrastive learning method in (Gao et al., 2021) but use relational corpus to train our sentence embedding model:

$$\mathcal{L}_i = -\log \frac{e^{sim(\mathbf{h}_i, \mathbf{h}_i^{+})/\tau}}{\sum_{j=1}^{N} (e^{sim(\mathbf{h}_i, \mathbf{h}_i^{+})/\tau} + e^{sim(\mathbf{h}_i, \mathbf{h}_i^{-})/\tau})} \tag{4}$$

where τ is a temperature hyperparameter and $\mathbf{h}$ represents the semantic embedding of the sentence. The cosine similarity is used as the similarity function: $sim(\mathbf{h}_1, \mathbf{h}_2) = \frac{\mathbf{h}_1^{\mathrm{T}} \mathbf{h}_2}{||\mathbf{h}_1|| \cdot ||\mathbf{h}_2||}$.

Finally, to identify the most similar sentences from the training pool for each test sentence, we calculate the cosine similarity score between the test sentence q and the target sentence s after encoding the sentence using our relational semantic sentence embedding model f_{sem}:

$$S_{sem}(q, s) = sim(f_{sem}(q), f_{sem}(s)). \tag{5}$$

3.2 Syntax-based Sentence Selection

3.2.1 Syntactic Representation

To acquire syntactic information for sentences, we employ Dependency Parsing technology. Dependency parsing is a natural language processing technique focused on analyzing the syntactic structure of a sentence by establishing relationships between words (Zhang et al., 2020a,b). Its objective is to uncover dependencies among words and finally outputs a dependency tree where words are nodes and dependencies are edges.

For sentence classification tasks, we directly utilize the entire dependency tree as the syntactic representation of the sentence. However, since the goal of Relation Extraction task is to identify the relationship between two predefined entities in a sentence, we further propose using a pruned dependency tree as the syntactic representation for sentences. Specifically, we first employ the shortest dependency path, preserving all nodes along the shortest dependency path connecting the two entities. And we propose a variant of extending the shortest dependency path. After obtaining the shortest dependency path, we add all nodes directly connected to the nodes in the shortest dependency path, achieving a balance between the entire tree and the SDP.

3.2.2 Syntactic Similarity

For evaluating syntactic similarity, we introduce the Tree Edit Distance (Zhang and Shasha, 1989; McCaffery and Nederhof, 2016; Lin et al., 2023) algorithm to calculate the edit distance between two syntax trees. Tree edit distance is a metric used to measure the similarity between two tree structures by quantifying the minimum number of operations required to transform one tree into another. These operations typically include insertion, deletion, and substitution of nodes and edges. The concept is analogous to the Levenshtein distance (Levenshtein, 1965) in strings but is specifically tailored for hierarchical structures like trees. By computing the minimum sequence of edit operations needed, the tree edit distance provides a quantitative measure of structural similarity:

$$ted(t_1, t_2) = 1 - \frac{EditDist(t_1, t_2)}{max(|t_1|, |t_2|)} \qquad (6)$$

where t is the dependency parsing tree of a sentence.

Finally, the syntactic similarity between two sentences is calculated by following function:

$$S_{syn}(q, s) = ted(f_{syn}(q), f_{syn}(s)) \qquad (7)$$

where f_{syn} is a dependency parser to produce the syntactic structure of a sentence.

3.3 Unified Semantic-Syntactic Sentence Selection

To incorporate semantics and syntax during sentence selection, we introduce a balancing parameter α to trade-off between semantic and syntactic similarities:

$$S(q, s) = \alpha S_{sem} + (1 - \alpha) S_{syn} \qquad (8)$$

4 Experiments

4.1 Experiment Setup

Datasets. We conduct our experiments on three datasets among two sentence classification tasks:
SemEval: a relation extraction task comes from SemEval-2010 Task 8 (Hendrickx et al., 2010), including about 10k human-annotated sentences covering 9 bidirectional relations and 1 special “NA” relation which means none of the above relations.
Re-TACRED: a revised version of the relation extraction dataset TACRED (Zhang et al., 2017) proposed by (Stoica et al., 2021). There are about 90k sentences for 39 common relations and 1 special “NA” relation.
TREC: a dataset for Question Classification task, containing about 6k annotated questions for 6 classes.

Metrics. For evaluation, we follow the standard metrics used in previous work: Micro-F1 metric (F1) for SemEval and Re-TACRED, and Accuracy (Acc.) for TREC.

Training Details To train our relational semantic sentence embedding model, we take KELM (Agarwal et al., 2021) corpus as our training data source. KELM contains about 15M sentences synthetically generated using a fine-tuned T5 model to cover 1522 relations from Wikidata. In order to construct both positive and negative data for training the contrastive model, we add a simple data pre-processing such as max length limitation and extract relations with more than 1,100 sentences in the KELM corpus.

As for the hyper-parameters in training our relational semantic sentence embedding model, we follow the setting in Gao et al. (2021).

split	# relations	# sentences per relation
train	326	1,000
val	326	50
test	326	50

Table 2: Statistics of relational data from KELM.

When obtaining the dependency parsing structures of sentences, we adopt the RoBERTa version of the off-the-shelf dependency parser SuPar[1] for its fast speed and good performance.

ICL setting We conduct the ICL experiments on the open source LLM LLaMA2-7b-chat (LLaMA2 for abbreviation) by local deployment and the closed source LLM GPT-3.5-turbo-0125 (ChatGPT for abbreviation) by invoking the API. For prediction on LLaMA2, we follow previous work (Brown et al., 2020) to compute the sentence perplexity of the sequence concatenation of input and candidate answers and select the final prediction using the one with the lowest perplexity. As for prediction on ChatGPT, we take the generation mode to generate the candidate label with highest probability. In order to a fair comparison of the number of examples in demonstrations, we use a 16-shot setting for all experiments. However, we also analyze the influence of the different amount of demonstrations among 2, 4, 8, 16.

As for prompting the LLM, we take the follow instruction for conducting in-context learning using LLMs: For the prompt template for each task, we use the simple input-label pair mode: "Sentence: <sen> , Label: <label> .". Relation labels are verbalized to semantic expressions with handcraft templates as introduced in (Zhang et al., 2023).

4.2 Baselines

Random We employ **Random** as a baseline, which randomly selects examples from the training pool as demonstrations for each test case (Brown et al., 2020) .

Syntax As discussed in 3.2, we also propose to use tree edit distance as similarity metric to retrieve demonstrations, named as **TreeEdit**.

Semantics For semantic-based methods, we reimplement three approaches utilizing different sentence embedding models to select the most similar examples from the training pool: (1) **RoBERTa**, proposed by (Liu et al., 2022b), who use RoBERTa (Liu et al., 2019) as sentence embedding to calculate the similarity. (2)**Sent-Trans**, proposed by (Hongjin et al., 2022). The Sentence-Transformers model (Reimers and Gurevych, 2019) is used as the sentence encoder. (3) **SimCSE**, proposed by Wan et al. (2023), who adopts the SimCSE model (Gao et al., 2021) as sentence encoder. (4) **RelCSE**, our relational sentence embedding model.

Method	Type		Task		
	Sem	Syn	SE	Re-T	TC
LLaMA2-7B					
Random	✗	✗	26.8	30.5	63.0
TreeEdit	✗	✓	48.0	45.5	88.0
RoBERTa	✓	✗	51.3	44.1	92.2
Sent-Trans	✓	✗	58.5	35.8	92.2
SimCSE	✓	✗	55.4	37.8	88.4
RelCSE	✓	✗	62.0	46.0	92.2
SSSS(ours)	✓	✓	67.9	57.1	93.2
GPT-3.5-Turbo-0125					
Random	✗	✗	41.5	20.2	77.6
TreeEdit	✗	✓	51.0	35.7	88.4
RoBERTa	✓	✗	50.5	35.7	89.2
Sent-Trans	✓	✗	54.5	35.2	91.0
SimCSE	✓	✗	56.0	34.0	89.4
RelCSE	✓	✗	62.7	40.6	91.4
SSSS	✓	✓	71.1	54.3	92.6

Table 3: Main results. SE and Re-T are relation extraction tasks, standing for SemEval and Re-TACRED. TC is the classification dataset TREC.

As for a fair comparison, we select the RoBERTa-large version for all the sentence embedding models.

4.3 Main Results

Random vs. Syntax Table 3 clearly indicates that **Syntax** incorporating sentence dependency syntactic information significantly outperforms **Random**. This method exhibited a pronounced enhancement in performance, as evidenced by an increase of 21.2 and 15.0 in the F1 score on the sentence relationship classification tasks on SemEval and Re-TACRED, respectively. These results suggest that the grammatical structure of sentences is highly effective in the context of learning based on LLMs.

Syntax vs. Semantics As to the introduced **Syntax** baseline, we find its data selection not only

[1] https://github.com/yzhangcs/parser

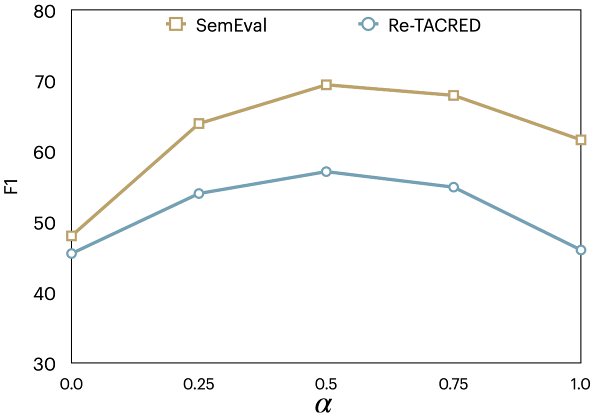


Figure 5: Final F1 scores with different α values.

outperforms **Random**, but also is very competitive to the **Semantics** method. This indicates the strong connections between relation extraction and dependency representations (He et al., 2018).

SimCSE vs. RelCSE With different training corpora, our RelCSE has consistent performance boost than simCSE. It is worth noting that in the two relation extraction datasets, RelCSE has 15.9 F1 score boost in average. As to the classification task, RelCSE also surpasses SimCSE on both the open-source LLaMA2-7B and the GPT-3.5.

Combining Semantics and Syntax Our proposed SSSS combines both the semantic and the syntactic representations. As shown in Table 3, our method outperforms other baselines and reaches the best results.

Scaling on LLMs Besides LLaMA2-7B, we test our method on GPT-3.5 to validate the effectiveness of the scaling properties. As presented in Table 3, our method has consistent performance gain on both the small open-source model and GPT-3.5. This verifies that the syntactic and the semantic representations could assist LLMs to select better fewshot exemplars in ICL.

4.4 Analysis of the Balance between Semantics and Syntax

As shown in Eqn 8 in Section 3.3, our method combines semantic representations and syntactic structures with a hyperparameter α. In order to analyze the influence of the value of α, we conduct experiments on our method among [0.0, 0.25, 0.5, 0.75, 1.0]. Specially, $\alpha = 0.0$ and $\alpha = 1.0$ are two special situations, which means the fully syntactic sentence selection and the fully semantic sentence selection, respectively.

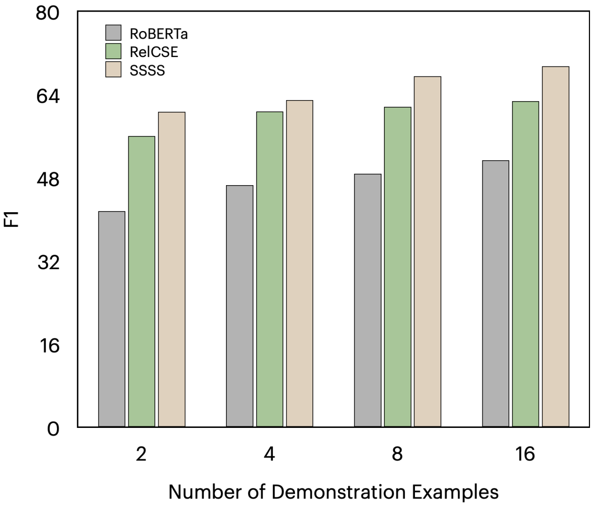


Figure 6: Results on SemEval with different number of demonstrations.

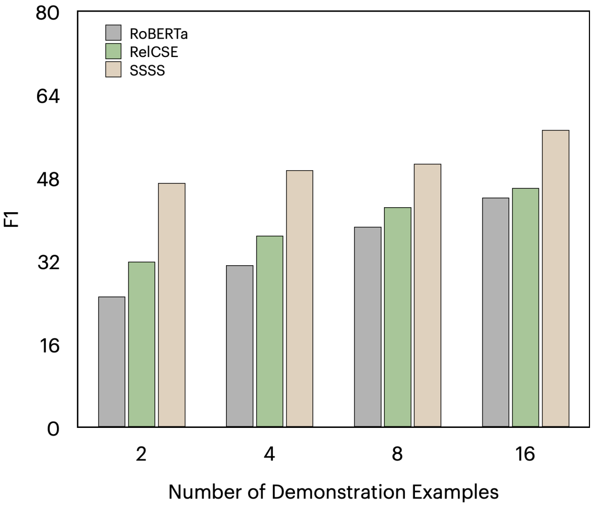


Figure 7: Results on Re-TACRED with different number of demonstrations.

From Figure 5, we find it is important to strike a balance between semantic and syntactic information, as an overly rigid reliance on syntax might lead to missing out on the flexibility and adaptability that semantic information provides. Combining both aspects can result in a more robust and contextually aware retrieval system.

4.5 Analysis of Demonstration Number and Order

The number and order of demonstration are also analyzed in many previous work when exploring ICL method. In this paper, we first conduct experiments with the demonstration number among [2, 4, 8, 16]. The results are shown in Figure 6 and 7. We find

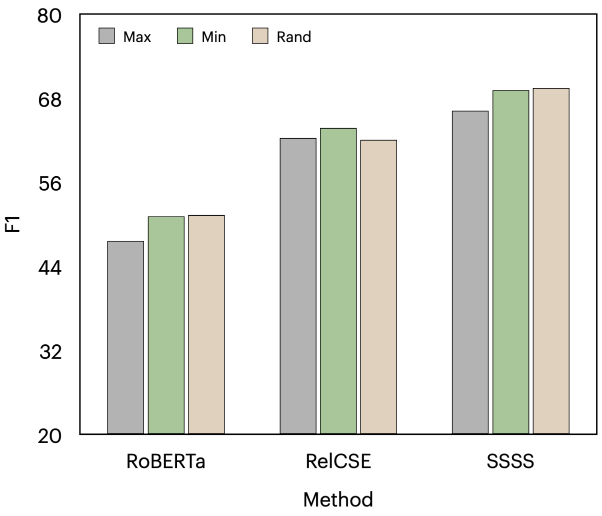


Figure 8: Results on SemEval with different demonstration orders.

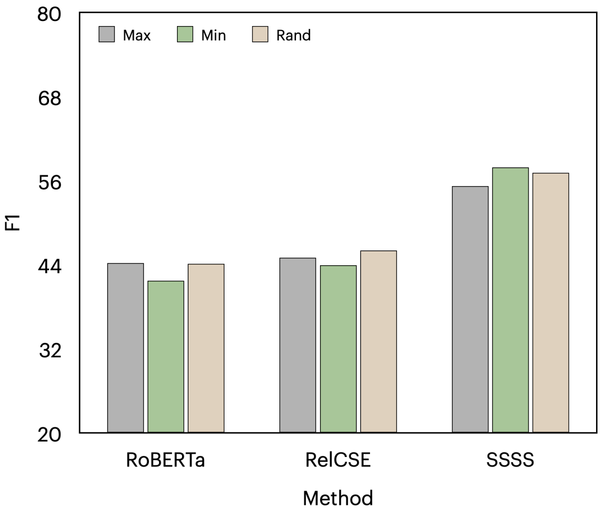


Figure 9: Results on Re-TACRED with different demonstration orders.

the two datasets have similar conclusions, where our proposed SSSS reaches the best results, and the performance grows up with more demonstrations.

Then, we further analyze the influence of the order in demonstration examples. We try three sequences: random sequence(Rand), most similar to least similar (Max) and least similar to most similar (Min). As shown in Figure 8 and 9, we find that our method is not sensitive to the order of demonstration examples, showing the method's robustness.

5 Related Work

LLM has exhibited strong ICL capabilities by understanding the task through learning from a small set of demonstration samples in the context (Dong et al., 2023; Luo et al., 2024). However, Zhao et al. (2021) reveals that ICL is unstable: the quality of LLM outputs is highly influenced by the design of prompts, and the selection and order of demonstration samples. In this paper, we focus on improving ICL from the perspective of enhancing the selection of demonstration samples.

Existing methods often rely on semantic-based approaches to find demonstration samples. Some researchers concentrate on finding samples with the highest similarity to the test input, where some common metrics include Euclidean or cosine distance (Liu et al., 2022a), mutual information (Sorensen et al., 2022), and BM25 (Rubin et al., 2022; Luo et al., 2023) are proven to be effective. Zhang et al. (2022) considers sample selection as a sequential decision problem and addresses it using a reinforcement learning-based algorithm. Wu et al. (2023) proposed a method to select samples that good at losslessly compressing testing samples based on the theory of compression. Li and Qiu (2023) proposed a new metric named as InfoScore, which utilizes LM feedback to assess the informativeness of a sample. Ye et al. (2023) trains a sample selection model that adopts the idea of Determinantal Point Processes (DPPs). Wang et al. (2023b) leverages LLM to select samples that closely match the latent topics of the test input. Iter et al. (2023) proposed a method based on cross-entropy difference for sample selection. In contrast to these semantic-based approaches, our work delves into sample selection from a syntactic perspective, presenting an orthogonal viewpoint.

6 Conclusion

In conclusion, based on our findings that ICL methods utilizing LLMs exhibit a bias towards sentences with similar syntactic structures, we propose a simple yet effective approach: Semantics and Syntax-based Sentence Selection (SSSS). Our approach seamlessly integrates semantic and syntactic considerations to enhance the selection of demonstrative sentences, thereby tackling a crucial aspect of ICL that has been previously underexplored. Experimental results on two sentence-level classification tasks across three datasets corroborate the effectiveness of our method.

Limitation

Scaling. Due to the computing resources, we only test our method on the open-sourced LLaMA2-7B and the commercial ChatGPT API. More experiments may be further conducted for verifying the effectiveness on larger open-source models.

Tasks scope. We test our method on relation extraction and classification tasks, while there are more task variations to explore, such as the pairwise textual entailment identifications.

References

Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. 2021. Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. In *North American Chapter of the Association for Computational Linguistics*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A survey on in-context learning.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Zhengqiu He, Wenliang Chen, Zhenghua Li, Meishan Zhang, Wei Zhang, and Min Zhang. 2018. See: Syntax-aware entity embedding for neural relation extraction. *ArXiv*, abs/1801.03603.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *SemEval*.

SU Hongjin, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, et al. 2022. Selective annotation makes language models better few-shot learners. In *The Eleventh International Conference on Learning Representations*.

Dan Iter, Reid Pryzant, Ruochen Xu, Shuohang Wang, Yang Liu, Yichong Xu, and Chenguang Zhu. 2023. In-context demonstration selection with cross entropy difference. pages 1150–1162, Singapore.

Vladimir Iosifovich Levenshtein. 1965. Binary codes capable of correcting deletions, insertions, and reversals. In *Doklady Akademii Nauk*, volume 163, pages 845–848. Russian Academy of Sciences.

Xiaonan Li and Xipeng Qiu. 2023. Finding support examples for in-context learning. pages 6219–6235, Singapore.

Boda Lin, Xinyi Zhou, Binghao Tang, Xiaocheng Gong, and Si Li. 2023. Chatgpt is a potential zero-shot dependency parser. *arXiv preprint arXiv:2310.16654*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022a. 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.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, William B Dolan, Lawrence Carin, and Weizhu Chen. 2022b. 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.

Yang Liu, Furu Wei, Sujian Li, Heng Ji, Ming Zhou, and Houfeng Wang. 2015. A dependency-based neural network for relation classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 285–290.

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

Man Luo, Xin Xu, Zhuyun Dai, Panupong Pasupat, Mehran Kazemi, Chitta Baral, Vaiva Imbrasaite, and Vincent Y Zhao. 2023. Dr.icl: Demonstration-retrieved in-context learning.

Man Luo, Xin Xu, Yue Liu, Panupong Pasupat, and Mehran Kazemi. 2024. In-context learning with retrieved demonstrations for language models: A survey.

Mingbo Ma, Liang Huang, Bowen Zhou, and Bing Xiang. 2015. Dependency-based convolutional neural networks for sentence embedding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 174–179, Beijing, China. Association for Computational Linguistics.

Martin McCaffery and Mark-Jan Nederhof. 2016. DTED: Evaluation of machine translation structure using dependency parsing and tree edit distance. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 491–498, Berlin, Germany. Association for Computational Linguistics.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. pages 2655–2671, Seattle, United States.

K Simonyan, A Vedaldi, and A Zisserman. 2014. Deep inside convolutional networks: visualising image classification models and saliency maps. In *Proceedings of the International Conference on Learning Representations (ICLR)*. ICLR.

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. pages 819–862, Dublin, Ireland.

George Stoica, Emmanouil Antonios Platanios, and Barnabás Póczos. 2021. Re-tacred: Addressing shortcomings of the tacred dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13843–13850.

Zhen Wan, Fei Cheng, Zhuoyuan Mao, Qianying Liu, Haiyue Song, Jiwei Li, and Sadao Kurohashi. 2023. Gpt-re: In-context learning for relation extraction using large language models. *arXiv preprint arXiv:2305.02105*.

Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. 2023a. Label words are anchors: An information flow perspective for understanding in-context learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9840–9855, Singapore. Association for Computational Linguistics.

Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. 2023b. Large language models are latent variable models: Explaining and finding good demonstrations for in-context learning.

Zhiyong Wu, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. 2023. Self-adaptive in-context learning: An information compression perspective for in-context example selection and ordering. pages 1423–1436, Toronto, Canada.

Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. 2023. Compositional exemplars for in-context learning.

Kai Zhang, Bernal Jiménez Gutiérrez, and Yu Su. 2023. Aligning instruction tasks unlocks large language models as zero-shot relation extractors. *arXiv preprint arXiv:2305.11159*.

Kaizhong Zhang and Dennis Shasha. 1989. Simple fast algorithms for the editing distance between trees and related problems. *SIAM journal on computing*, 18(6):1245–1262.

Yiming Zhang, Shi Feng, and Chenhao Tan. 2022. Active example selection for in-context learning. pages 9134–9148, Abu Dhabi, United Arab Emirates.

Yu Zhang, Zhenghua Li, and Zhang Min. 2020a. Efficient second-order TreeCRF for neural dependency parsing. In *Proceedings of ACL*, pages 3295–3305.

Yu Zhang, Houquan Zhou, and Zhenghua Li. 2020b. Fast and accurate neural CRF constituency parsing. In *Proceedings of IJCAI*, pages 4046–4053.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.