SCOI: Syntax-augmented Coverage-based In-context Example Selection for Machine Translation

Anonymous ACL submission

Abstract

In-context learning (ICL) greatly improves the performance of large language models (LLMs) on various down-stream tasks, where the improvement highly depends on the quality of demonstrations. In this work, we introduce syntactic knowledge to select better in-context examples for machine translation (MT). We propose a new strategy, namely **S**yntax-augmented **CO**verage-based **I**n-context example selection (SCOI), leveraging the deep syntactic structure beyond conventional word matching. Specifically, we measure the set-level syntactic coverage by computing the coverage of polynomial terms with the help of a simplified tree-to-polynomial algorithm, and lexical coverage using word overlap. Furthermore, we devise an alternate selection approach to combine both coverage measures, taking advantage of syntactic and lexical information. We conduct experiments with two multi-lingual LLMs on six translation directions. Empirical results show that our proposed SCOI obtains the highest average COMET score among all learning-free methods, indicating that combining syntactic and lexical coverage successfully helps to select better in-context examples for MT.

1 Introduction

In-context learning (ICL) has become a popular prompting strategy to elicit the power of large language models (LLMs) across a wide range of natural language processing (NLP) tasks (Brown et al., 2020; Min et al., 2022; Dong et al., 2023). In ICL, several demonstrations including both task input and ground truth output are presented in the input context, to make LLMs understand the specific down-stream task and produce better results.

The performance of ICL highly depends on the quality of in-context examples, and it is thus of great significance to explore selecting better examples for ICL (Rubin et al., 2022). There have been numerous works on in-context example selection for monolingual tasks like natural language inference, commonsense reasoning and semantic parsing (Li et al., 2023; Ye et al., 2023; Gupta et al., 2023; Liu et al., 2024). Unlike these tasks above, machine translation (MT) involves multiple languages and requires a more sophisticated design of in-context example selection. Recently, there have some attempts on in-context example selection specially for MT, which leverage word-level matching (Agrawal et al., 2023), embedding-based scoring (Moslem et al., 2023; Ji et al., 2024; Zhu et al., 2024) or combination of superficial features (Kumar et al., 2023).

In previous studies, for both statistical MT and neural MT, syntax plays a crucial role in improving model performance (Williams and Koehn, 2014; Wu et al., 2017). However, in case of ICL, most existing works focus on superficial features but pay little attention to the syntactic structure of sentences. To achieve a high translation quality, it requires not only an accurate word translation but also a proper syntactic structure of the generated target sentence. Hence, syntactic information should also play a big part in MT even in the era of LLMs.

Compared with independent selection, it has been proved that selecting in-context examples as an entire set based on the set-level coverage leads to a better diversity while reducing redundancy and avoiding sub-optimal results (Gupta et al., 2023). As a typical NLP task, MT would also benefit from in-context examples with a high set-level coverage. Therefore, beyond the conventional lexical coverage, high syntactic coverage is also necessary to select informative in-context examples for MT.

In this work, we propose **S**yntax-augmented **CO**verage-based **I**n-context example selection, SCOI [1], to boost LLMs' performance on MT. Specifically, to measure syntactic coverage, we first simplify a tree-to-polynomial algorithm (Liu

---

[1] /ˈskoʊɪ/.

et al., 2022), which is originally costly but has been reduced to no more than quadratic time complexity after simplification. Using this new algorithm, we convert syntax trees into polynomials and then compute the set-level syntactic coverage based on vector representations of polynomial terms. Meanwhile, we compute the proportion of word overlap to measure set-level lexical coverage. After that, we design an alternate approach to combine both coverage measures, so that word-level and syntax-level features would complement each other.

We evaluate SCOI on 6 translation directions (German, French, Russian into and out of English) based on two open-source multi-lingual LLMs, $XGLM_{7.5B}$ (Lin et al., 2022) and Alpaca (Taori et al., 2023). Among all learning-free methods, SCOI obtains the highest COMET scores on 4 out of 6 translation directions and the highest average COMET score. Especially, on Russian-to-English and English-to-Russian translations, SCOI even outperforms the learning-based CTQ Scorer (Kumar et al., 2023) when using Alpaca.

Our contributions can be summarized as follows:

- Going beyond superficial word matching, we introduce the knowledge of syntactic structure to in-context example selection for MT.

- To take advantage of both word overlap and syntactic resemblance, we propose a novel framework to ensure a high set coverage at both word and syntax level for in-context example selection, and empirical experiments validate the effectiveness of our method.

- We design a simplified tree-to-polynomial algorithm owning a complexity upper bound of no more than quadratic time. In contrast, that of the original version could be polynomial time with an arbitrarily large degree.

Our code will be publicly available after the publication of this paper.

2 Related Work

Prompting LLMs for better performance has been one of the mainstream trends of NLP research. There have been a large number of studies on prompting strategies for MT in recent years (Vilar et al., 2023; Zhang et al., 2023). Puduppully et al. (2023) decompose the translation process into a sequence of word chunk translations to improve LLMs' performance on translation between linguistically related languages. Ghazvininejad et al. (2023) propose to present LLMs with a set of possible translations for a subset of the input words from bilingual dictionaries to improve LLMs' performance on low-resource and out-of-domain MT. He et al. (2024) prompt LLMs with selected knowledge including keyword pairs, topics and sentence pairs to emulate human-like translation. Zhang et al. (2024) manage to teach LLMs an unseen language on the fly with the help of a small parallel corpus and a dictionary. Guo et al. (2024) first create a textbook including vocabulary list, language examples with syntax patterns and translate instructions using LLMs and then prompt LLMs with the textbook just created to better translate low-resource languages. Zhu et al. (2024) prompt LLMs with both sentence-level and word-level demonstrations, the former selected with a margin-based score and the latter being word pairs most related to the test input appeared in the former.

Among various prompting strategies, ICL plays a key role. Rubin et al. (2022) suggest that the performance of ICL strongly depends on the selected in-context examples. Thus it is of great significance to select better examples using various strategies. Li et al. (2023) propose to train a unified demonstration retriever for ICL across a wide range of tasks. Ye et al. (2023) make use of determinantal point processes (DPPs) to ensure both relevance and diversity of examples. Liu et al. (2024) select examples in a sequential rather than "select then organize" way that leverages the LLM's feedback on varying context, aiding in capturing inter-relationships and sequential information among examples. Gupta et al. (2023) define measure of set-level information coverage and select examples based on it, which inspires our work.

There are some example selection strategies customized for MT. Agrawal et al. (2023) select examples based on n-gram overlap. Moslem et al. (2023) select examples based on sentence embedding similarity. Kumar et al. (2023) train language-specific regression models to combine various features for example selection. Ji et al. (2024) select examples based on submodular functions combining surface/semantic similarity and diversity within examples. To the best of our knowledge, no previous work has made use of syntactic information in in-context example selection for MT.
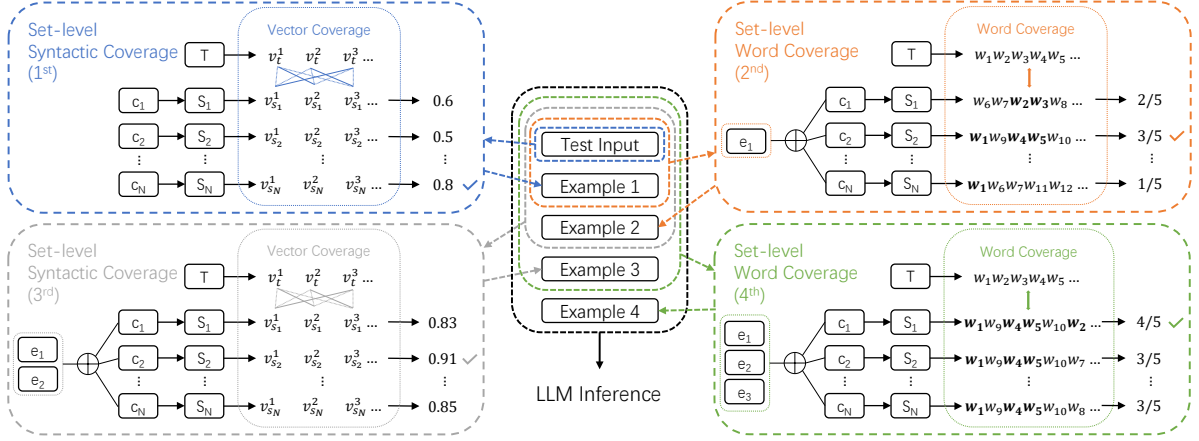
Figure 1: Overview of SCOI. Each example is selected based on how well the test input is covered by the current candidate plus the existing examples selected in previous steps at syntax level and word level alternately. In each step, T, e_i, $\oplus$, c_i, S_i denote the test input, the i-th selected example, concatenation of selected examples and one candidate, the i-th candidate from the example database, the to-be-scored set including the selected examples plus the i-th candidate, respectively.

3 Method

We propose to select in-context examples based on both syntactic and lexical coverage to better apply LLMs for MT. Specifically, to measure the set-level syntactic coverage, we first simplify a tree-to-polynomial algorithm, making it practical to run on large MT datasets, and then compute the coverage of vector representations of polynomial terms. To measure the set-level lexical coverage, we simply consider the proportion of word overlap. After that, we design an alternate strategy to take advatage of both lexical and syntactic knowledge. An overview of our proposed method, SCOI, is presented in Figure 1.

3.1 Polynomial Representation of Syntactic Structure

Liu et al. (2022) convert dependency trees into polynomials recursively and compute the distance between polynomials to measure the syntactic similarity between sentences from different languages. Specifically, given the number of dependency labels d, dependency trees will be transformed into polynomials based on two variable sets: $X = \{x_1, x_2...x_d\}$ and $Y = \{y_1, y_2, ...y_d\}$. Considering a leaf node with label l as n^l, its corresponding polynomial is $P(n^l, X, Y) = x_l$. For a non-leaf node m^l with label l, its polynomial is:

$$P(m^l, X, Y) = y_l + \prod_{i=1}^{k} P(n_i, X, Y), \quad (1)$$

where $n_1, ..., n_k$ are all child nodes of m^l.

However, the algorithm can be of very high complexity when the dependency tree is large. In MT, there are often millions of data to be processed and it is thus impractical to make use of the original algorithm from Liu et al. (2022). Therefore, we propose a simplified polynomial algorithm, reducing the complexity of tree-to-polynomial conversion to no more than quadratic time.

Concretely, our newly defined polynomial is based on only one variable set $X = \{x_1, x_2...x_d\}$. For a leaf node n^l, its polynomial remains $P(n^l, X) = x_l$. For a non-leaf node m^l with child nodes $n_1, ..., n_k$, its polynomial is:

$$P(m^l, X) = x_l \cdot (1 + \sum_{i=1}^{k} P(n_i, X)). \quad (2)$$

where each term $x_1^{e_{x_1}} x_2^{e_{x_2}} ... x_d^{e_{x_d}}$ in the polynomial corresponds to a path from the root node to one certain node in the tree, and e_{x_i} indicates the number of nodes with the i-th dependency label on that path. Given a sentence with a dependency tree rooted in Node r, the polynomial representing the syntactic structure of that sentence is $P(r, X)$.

We analyze the complexity of the original and our simplified tree-to-polynomial algorithms in Appendix A.

3.2 Measure of Set-level Syntactic Coverage

Given a test input x and a set of in-context examples Z, the set of salient aspects (e.g., entities, keywords, etc.) of x being S_x, the set-level information coverage of in-context examples is defined

3

as (Gupta et al., 2023):

$$\texttt{SetCov}(x, Z) = \sum_{s \in S_x} \max_{z \in Z} \texttt{c}(s, z), \quad (3)$$

where $\texttt{c}(s, z)$ measures the coverage or recall of a single salient aspect s by example z.

For better parallelization and to better fit the salient aspects denoting syntax in this work, which are vector representations of polynomial terms from the tree-to-polynomial algorithm, we reformulate Equation 3 to the set-level syntactic coverage:

$$\texttt{SynSetCov}(x, Z) = \frac{1}{|T_x|} \sum_{s \in T_x} \max_{t \in T_Z} \texttt{c}(s, t), \quad (4)$$

where T_x is the multiset [2] of terms in the polynomial representation of the dependency tree of x, $T_Z = \bigcup_{z \in Z} T_z$ is the multiset of all the terms in polynomials of dependency trees of all the in-context examples in Z, s and t denote terms in T_x and T_Z respectively, and $\texttt{c}(s, t)$ computes the similarity of term s and term t.

To compute $\texttt{c}(s, t)$, we first compute the distance between the two polynomial terms. Note that a term $t = x_1^{e_{x_1}} x_2^{e_{x_2}} ... x_d^{e_{x_d}}$ can be written as a term vector with d entries:

$$v_t = [e_{x_1}, e_{x_2}, ..., e_{x_d}], \quad (5)$$

where each entry represents the exponent of the corresponding variable. The distance between terms s and t can thus be calculated by the Manhattan distance (Craw, 2017) between vectors v_s and v_t:

$$\texttt{d}(s, t) = \| v_s - v_t \|_1. \quad (6)$$

As distance is negatively correlated with similarity, we compute $\texttt{c}(s, t)$ using the normalized distance:

$$\texttt{c}(s, t) = \frac{1}{1 + \texttt{d}(s, t)}. \quad (7)$$

In this way, $\texttt{c}(s, t)$ is a normalized value between 0 and 1. Note that each term in the polynomial represents a root-to-node path in the tree. So $\texttt{SynSetCov}(x, Z)$ indicates the average coverage of each path in the dependency tree of x by all the dependency trees in Z.

------

[2] Since we take repeated elements into account, we use *multiset* (Hickman, 1980) that allows repetition of elements instead of *set* in this work.

Algorithm 1 Greedy Optimization of Set Coverage

Require: Example database $\mathcal{T}$; test input x; desired number of demonstrations k; coverage scoring function SynSetCov and WordSetCov.
1: $Z \leftarrow \emptyset$ ▷ Selected in-context examples.
2: $Z_{\text{curr}} \leftarrow \emptyset$ ▷ Current set cover.
3: $\texttt{curr\_syn\_cov} \leftarrow -\inf$
4: $\texttt{curr\_word\_cov} \leftarrow -\inf$
5: **while** $|Z| < k$ **do**
6:    **if** $|Z| \equiv 0 \pmod 2$ **then** ▷ Odd-numbered to-be-selected example.
7:      $z^*, \texttt{next\_syn\_cov} = \underset{z \in \mathcal{T} - Z}{\arg\max} \texttt{SynSetCov}(x, Z_{\text{curr}} \cup z)$
8:      **if** $\texttt{next\_syn\_cov} > \texttt{curr\_syn\_cov}$ **then**   ▷ Pick z^*.
9:        $\texttt{curr\_syn\_cov} \leftarrow \texttt{next\_syn\_cov}$
10:        $Z \leftarrow Z \cup z^*$
11:        $Z_{\text{curr}} \leftarrow Z_{\text{curr}} \cup z^*$
12:      **else**   ▷ Start a new one if no increase.
13:        $Z_{\text{curr}} \leftarrow \emptyset, \ \texttt{curr\_syn\_cov} \leftarrow -\inf$
14:      **end if**
15:    **else**   ▷ Even-numbered to-be-selected example.
16:      $z^*, \texttt{next\_word\_cov} = \underset{z \in \mathcal{T} - Z}{\arg\max} \texttt{WordSetCov}(x, Z_{\text{curr}} \cup z)$
17:      **if** $\texttt{next\_word\_cov} > \texttt{curr\_word\_cov}$ **then**   ▷ Pick z^*.
18:        $\texttt{curr\_word\_cov} \leftarrow \texttt{next\_word\_cov}$
19:        $Z \leftarrow Z \cup z^*$
20:        $Z_{\text{curr}} \leftarrow Z_{\text{curr}} \cup z^*$
21:      **else**   ▷ Start a new one if no increase.
22:        $Z_{\text{curr}} \leftarrow \emptyset, \ \texttt{curr\_word\_cov} \leftarrow -\inf$
23:      **end if**
24:    **end if**
25: **end while**
26: **return** Z

3.3 Measure of Set-level Lexical Coverage

In this work, we simply measure the set-level lexical coverage by computing the proportion of word overlap:

$$\texttt{WordSetCov}(x, Z) = \frac{|W_x \cap W_Z|}{|W_x|}, \quad (8)$$

where W_x is the multiset of the words in x and $W_Z = \bigcup_{z \in Z} W_z$ is the multiset of all the words in all the examples in Z.

3.4 Combined Syntactic and Lexical Coverage

Combining syntax-level and word-level coverage could make them complement each other and thus help select better in-context examples for MT. In this work, we propose an alternate way to combine both.

For convenience, we number ICL examples starting from 1. Specifically, for each odd-numbered example, we select it based on how well the current candidate, along with the existing examples selected in previous steps, covers the test input in syntax, while for each even-numbered example, we select it based on set-level lexical coverage. To put it more concretely, we select the first example with the highest set-level (only the first example at this time) syntactic coverage and the second example with the highest set-level (including the first and the second example) lexical coverage.

Following Gupta et al. (2023), we use a greedy algorithm to select the optimal set as shown in Al-

4

gorithm 1. It alternately selects examples that lead to the maximum syntactic coverage (lines 7-11) and lexical coverage (lines 16-20). If no example brings further increase in coverage, the algorithm reserves the selected examples and starts another round (lines 12-13 and 21-22).

4 Experimental Setup

We follow Kumar et al. (2023) to set up our experiments.

4.1 Datasets and Evaluation Metrics

Language	ISO Code	Dataset	#Pairs (M)
German	DE	Europarl	1.83
French	FR	Europarl	1.92
Russian	RU	ParaCrawl	5.38

Table 1: Data statistics.

Test Set We perform our evaluation on the *devtest* set of FLORES-101 (Goyal et al., 2022), which has 1012 sentences with translations in 101 languages. We experiment between English and 3 common languages including German, French and Russian.

Example Database We use Europarl (Koehn, 2005) for German and French and ParaCrawl (Bañón et al., 2020) for Russian as example database. Detailed statistics are shown in Table 1.

Evaluation Metrics We report COMET (Rei et al., 2020) scores from `wmt20-comet-da`, which is considered a superior metric for MT nowadays (Kocmi et al., 2021). We report BLEU scores from sacreBLEU (Post, 2018) in Appendix B.

4.2 Pre-processing

We parse all the datasets with spaCy (Honnibal et al., 2020) to get dependency trees for our syntax-based approaches. The spaCy models we use for different languages are listed in Appendix C.

We use Sacremoses [3] to tokenize all the languages for the lexical coverage computation.

4.3 Large Language Models

XGLM$_{7.5B}$ (Lin et al., 2022) and Alpaca (Taori et al., 2023) are used in our experiments. XGLM is a multilingual generative language model supporting 30 languages and has 7.5B parameters

---

[3] https://github.com/hplt-project/sacremoses

---

in total. Alpaca is a 7B model fine-tuned from LLaMA (Touvron et al., 2023) on 52K instruction-following data.

4.4 Prompt Template

The number of in-context examples is set to 4 in our experiments. We use the same prompt template used in Kumar et al. (2023) for XGLM and that in He et al. (2024) for Alpaca. Please refer to Appendix D for details.

Other details of our implementation can be found in Appendix E.

4.5 Baselines

Random: Examples are selected randomly for each test input from the example database. We report the average result of 3 different random seeds.

BM25: We use the BM25 algorithm implemented by Bassani (2023) to retrieve the top-k matching examples in the example database for each test input.

Following Agrawal et al. (2023) and Kumar et al. (2023), all the compared methods below re-rank examples based on top-100 examples retrieved by BM25 for each test input.

R-BM25: We evaluate R-BM25 (Agrawal et al., 2023) for comparison, which ensures n-gram coverage and diversity.

Fuzzy: We evaluate Fuzzy (Moslem et al., 2023), where examples that are most similar in sentence-level embedding are selected. We use sentence transformers (Reimers and Gurevych, 2019) with `paraphrase-multilingual-MiniLM-L12-v2` (Reimers and Gurevych, 2020) to reimplement it.

CTQ Scorer: We evaluate CTQ Scorer (Kumar et al., 2023) for comparison, which is a learning-based method combining multiple features including number of tokens, similarity in LaBSE embeddings (Feng et al., 2022), perplexity, etc. It trains a specific regression model for each language pair.

SCOI: Our proposed method described in Section 3.

5 Results and Analysis

5.1 Main Results

Main results are shown in Table 2. SCOI obtains the highest COMET scores of 4 out of 6 translation directions and the highest average COMET score among all learning-free methods using both

System	Into EN			Out of EN			Avg.
	DE	FR	RU	DE	FR	RU	
XGLM							
Learning-free							
Random	63.53	70.80	53.41	43.03	53.23	42.70	54.45
BM25	63.21	71.36	52.48	44.13	**55.54**	44.58	55.22
R-BM25	64.13	71.18	54.06	44.83	55.21	45.92	55.89
Fuzzy	64.40	**71.92**	53.37	44.45	55.23	44.69	55.68
SCOI (*ours*)	**64.67**	71.26	**54.08**	**44.87**	55.31	**46.47**	**56.11**
Learning-based							
CTQ Scorer	65.38	70.65	53.48	45.52	56.00	48.59	56.60
Alpaca							
Learning-free							
Random	69.71	76.64	57.47	42.60	56.58	28.61	55.27
BM25	69.08	76.41	58.52	43.65	57.34	32.63	56.27
R-BM25	69.71	**76.70**	57.69	43.87	**59.17**	34.78	56.99
Fuzzy	69.72	76.36	58.12	**44.10**	57.25	30.57	56.02
SCOI (*ours*)	**69.79**	76.08	**58.66**	**44.10**	57.97	**36.26**	**57.14**
Learning-based							
CTQ Scorer	70.39	76.57	58.63	45.55	58.71	35.68	57.59

Table 2: COMET scores of 4-shot ICL performance of SCOI and other methods for translation on all 6 directions of 2 language models. All the methods except CTQ Scorer are learning-free, which do not require task, language or LLM-specific training. "**Avg.**" refers to the average score across all 6 directions. The highest scores among learning-free methods are in **bold** text.

XGLM and Alpaca, showing competitive performance across language models. Using XGLM, SCOI outperforms the learning-based CTQ Scorer on "RU-EN", while using Alpaca, SCOI even outperforms CTQ Scorer on both "RU-EN" and "EN-RU". Note that Alpaca seems not good at generating Russian, which can be inferred from its poor random and BM25 baselines. But SCOI greatly improves its performance on "EN-RU" and shows amazing ability in teaching an LLM to better translate into a language that appeared less during training.

We observe that SCOI shows obvious preference across different languages. For example, it fails to benefit "FR-EN" but improves performance on "RU-EN". Besides different natures of different languages, this might be also due to different capabilities of syntax parsers for different languages. We find that when the parser performs poorly (e.g., the French parser), SCOI also performs less competitively, while a more powerful parser (e.g., the Russian one) leads to better performance of SCOI. Details of the relation between parser and SCOI's performance can be found in Appendix F.

We also experiment on GPT-3.5 (Ouyang et al., 2022), which is an API-based LLM. Results are presented in Appendix G.

5.2 Ablation Study

Method	Into EN			Out of EN			Avg.
	DE	FR	RU	DE	FR	RU	
SCOI	64.67	71.26	54.08	**44.87**	55.31	46.47	**56.11**
w/o syntax	64.44	**71.52**	53.33	43.99	55.52	46.22	55.84
w/o word	63.84	70.95	53.30	42.55	**56.25**	**46.82**	55.62

Table 3: Ablation results of SCOI on XGLM. "w/o syntax" refers to select using word-level coverage only and "w/o word" refers to select using syntax-level coverage only.

To explore the effect of syntactic and lexical information, we perform ablation experiments using XGLM. Since SCOI uses both syntactic and lexical coverage, we evaluate the syntactic coverage-only and lexical coverage-only selection methods.

As shown in Table 3, either word-only or syntax-only coverage has limitations on some translation directions. For instance, the syntax-coverage method performs poorly on "FR-EN" and "EN-DE" while the word-coverage one performs less competitively on "RU-EN" and "EN-DE". With the help of alternate word-coverage and syntax-coverage, our proposed method of combined coverage makes the best of both worlds by and large, performing satisfactorily on all directions except "EN-FR" and achieves the highest average score.

5.3 Experiments with Different Selection Modes

Mode	Into EN			Out of EN			Avg.
	DE	FR	RU	DE	FR	RU	
BM25	63.21	**71.36**	52.48	44.13	55.54	44.58	55.22
Top-k	64.15	70.79	53.71	43.22	54.75	**46.49**	55.52
DPP	63.64	70.71	53.65	43.61	**55.55**	45.48	55.44
SCOI	**64.67**	71.26	**54.08**	**44.87**	55.31	46.47	**56.11**

Table 4: COMET scores of 4-shot ICL performance on XGLM of different selection modes, all trying to make use of syntactic information.

We explore different modes of in-context example selection including top-k, DPP and our proposed coverage-based SCOI using XGLM to see how to make the most of syntactic information in in-context example selection.

Top-k We select the top-k examples with the highest syntactic similarity based on the polynomial distance used in Liu et al. (2022) for each test input from the example database. Details of polynomial distance can be found in Appendix H.

DPP Inspired by Ye et al. (2023) and Yang et al. (2023), we explore selecting in-context examples for MT using Determinantal Point Processes (DPPs). DPPs are elegant probabilistic models capable of selecting a representative subset from a larger, potentially redundant set.

To incorporate both lexical diversity (differences in vocabulary coverage between different examples) and syntactic relevance (similarity between the candidate example and the test input) in the in-context example selection process, we utilize the same equation that combines diversity and relevance as used in Ye et al. (2023):

$$\log \det(\mathbf{L}'_S) = \frac{1}{\lambda} \sum_{i \in S} r_i + \log \det(\mathbf{L}_S), \quad (9)$$

where r_i represents syntactic relevance, measured by the normalized polynomial distance between each candidate example and the test input, and $\mathbf{L}_S$ denotes lexical diversity, constructed through the dot product of word vectors of all candidate examples.

Given the $\log \det(\mathbf{L}'_S)$, we can select the representative subset S_{best} of size k as follows:

$$S_{\text{best}} = \underset{S \subseteq Z, |S|=k}{\arg\max} \det(\mathbf{L}'_S). \quad (10)$$

For the actual selection of S_{best}, we utilize the exact implementation of the greedy algorithm from Ye et al. (2023), originally proposed in Chen et al. (2018). Other details of DPP are presented in Appendix I.

Results Results are shown in Table 4. Note that all the methods re-rank on the basis of top-100 examples of each test input retrieved by BM25. Thus BM25 is a comparable baseline.

The top-k mode does achieve a slightly higher average score compared with BM25 but in fact shows some performance drop on "FR-EN", "EN-DE" and "EN-FR" directions. This indicates simply re-ranking based on only syntactic closeness cannot necessarily secure improvement.

The DPP mode shows a slight improvement on average, but its performance fluctuates across translation directions. This indicates that simply incorporating syntax similarity into the relevance term in DPP does not necessarily yield desired improvement and how to effectively combine lexical and syntactic information using DPPs still requires exploration, which we leave for future work.

SCOI, however, performs better compared with the baselines above, obtaining highest or competitive scores across all 6 translation directions and getting the highest average score. This proves that selecting examples based on syntactic and lexical coverage alternately effectively leverages syntactic information and leads to better ICL performance.

Order	Into EN			Out of EN			Avg.
	DE	FR	RU	DE	FR	RU	
Word First	64.45	70.64	53.76	**45.39**	**55.91**	45.63	55.96
Syntax First	**64.67**	**71.26**	**54.08**	44.87	55.31	**46.47**	**56.11**

Table 5: COMET scores of 4-shot ICL performance on XGLM of different orders of alternating.

5.4 Analysis on the Selection Order

In this section, we analyze the effect of the order of alternating during the selection of SCOI.

By default, the order of alternating is syntax-first, i.e., selecting odd-numbered examples using syntactic coverage and even-numbered ones using lexical coverage. We experiment on a reversed order (i.e., word-first) for comparison.

Experimental results on XGLM are shown in Table 5. On average, the syntax-first order is slightly better than the word-first one. This indicates that focusing on syntax first can organize a better set of in-context examples.

7

	DE	EN
Input & Gold	Nach einer Woche voller Verluste in der Zwischenwahl erzählte Bush dem Publikum von **der Ausweitung des Handels in Asien**.	After a week of losses in the midterm election, Bush told an audience about **the expansion of trade in Asia**.
BM25 Prediction	-	After a week of losses in the mid-election campaign, President Bush told his audience **that trade in Asia had been expanded**.
Example-1	Deshalb geht meiner Ansicht nach der Verlust von Sprachen mit dem Verlust von Lebensweisen einher.	I think, therefore, that if we lose languages we lose forms of life.
Example-2	Ich stimme mit dem Standpunkt der Berichterstatterin überein und bin mit den eingeführten Veränderungen, wie der Ausweitung der Mindestdauer des Mutterschaftsurlaubs von 14 auf 20 Wochen, dem Grundsatz einer Bezahlung in voller Höhe des bisherigen Einkommens, der Einführung von Gesundheitsschutzbestimmungen am Arbeitsplatz und dem Verbot der Kündigung, einverstanden.	I agree with the position of the rapporteur and with the changes introduced, such as the extension of the minimum period for maternity leave from 14 to 20 weeks, the principle of pay equivalent to complete earnings, the establishment of health and safety requirements in the workplace, and the prohibition of dismissal.
Example-3	Es muss eine grundlegende Strategie sein, die alle Ursachen der Krise einbezieht: die Veränderung der Ernährungsgewohnheiten in Asien, die rasche Ausweitung des Anbaus von Biokraftstoffen usw.	It must be a basic strategy that tackles all the causes of the crisis: changing food habits in Asia, the rapid rise in the cultivation of biofuels, etc.
Example-4	Das hat seinen Widerhall bei seinem Publikum gefunden, von dem in dieser Woche 50.000 die Online-Petition für seine Freilassung unterzeichnet haben.	This has resonated among his audience, 50 000 of whom have this week signed the online petition asking for his release.
SCOI Prediction	-	After a week of losses in the mid-term election, Bush told the audience about **the expansion of trade in Asia**.

Table 6: An end-to-end German-to-English translation example. "Input & Gold" refers to the test input and the gold reference. "BM25 Prediction" refers to XGLM's prediction given the test input and examples selected by BM25, which are shown in Appendix K. "Example-i" refers to the i-th example selected by SCOI. "SCOI Prediction" shows the predict of XGLM given the test input and the 4 in-context examples selected by SCOI.

5.5 Case Analysis

An end-to-end German-to-English case is presented in Table 6, showing the test input, ground truth, selected examples of SCOI and model prediction of XGLM with in-context examples selected by BM25 and SCOI separately. The set of in-context examples selected by SCOI brings a good demonstration at both syntax level and word level. For instance, the first example, which is selected based on syntactic coverage, shows very close syntactic structure to the test input, with multiple prepositional phrases ("meiner Ansicht nach", "von Sprachen", "mit dem Verlust von Lebensweisen"), a very alike structure of main clause (a verb and a noun phrase) and similarly complex noun phrases ("der Verlust von Sprachen mit dem Verlust von Lebensweisen"). The second example, which is selected based on lexical coverage, covers many words as expected ("einer", "voller", "Ausweitung", etc.). The third example, again selected based on syntactic coverage, again shows very homologous syntactic structure including use of multiple prepositional phrases, complex noun phrases and similar main clause. The fourth example, based on lexical coverage, covers some other important words ("Publikum", "Woche", etc).

Table 6 also compares SCOI's system output with that of BM25. BM25 fails to construct the proper syntactic structure when translating the German phrase "der Ausweitung des Handels in Asien" and turns it into a reported clause "that trade in Asia had been expanded", thus losing accuracy. Note that "der Ausweitung des Handels in Asien" (the expansion of trade in Asia) does not include temporal information and it could be a bygone, a current state or a future trend, while the result of BM25 assumes that it is something that happened in the past, which is inconsistent with the original meaning of the input sentence. However, SCOI, combining syntactic and lexical coverage, is able to output the exact noun phrase "the expansion of trade in Asia", which is consistent with the syntactic structure in the source German sentence and much more accurate in translation. For the complete end-to-end case of BM25, please refer to Appendix K.

6 Conclusion

In this work, we introduce syntactic information to in-context example selection for MT. We propose to measure set-level syntactic coverage with the set-level coverage of polynomial terms representing root-to-node paths in syntax trees based on a simplified tree-to-polynomial algorithm and select in-context example selection for MT using an alternate mode where syntactic and lexical coverage are considered alternately to combine information of syntax and word. Our proposed method obtains the highest average COMET score among all learning-free methods, indicating that combining syntactic and lexical coverage during in-context example selection is helpful for MT. We call on the NLP community to pay more attention to syntactic knowledge for syntax-rich tasks like MT.

Limitations

Syntax Parser Our syntax-based method is based on reliable parsers and might not work well for low-resource languages. Meanwhile, dependency parsing could be costly when dealing with large datasets, which makes SCOI more time-consuming in such situations.

Semantics We have not tried semantic information (e.g., sentence embeddings) in our method.

Word-level Coverage We have not tried other advanced word-level coverage methods (e.g., weighted words based on their frequencies or n-gram features).

The Original Tree-to-Polynomial Algorithm Due to limited time, we have not completed the evaluation of the original tree-to-polynomial algorithm on our method to compare with our simplified version. In fact, the algorithm got stuck at a long sentence with a large dependency tree and failed to finish that instance before we killed the process due to overlong running time.

The Simplified Tree-to-Polynomial Algorithm There might be some information loss in the simplified tree-to-polynomial algorithm. For example, each term in the polynomial only presents the number of each dependency label on its corresponding root-to-node path but cannot show the exact order of these labels. In other words, our simplified tree-to-polynomial algorithm is a many-to-one mapping and is thus irreversible.

Ethics Statement

Task	Time (min)
BM25 Pre-selection	12
Dependency Parsing	60
Tokenization	4
Combined Coverage	9
LLM Inference	90

Table 7: Average computation time on German into/out of English using XGLM.

Computational Budget We run pre-processing and in-context example selection on Intel® Xeon® Gold 6348 CPU and the LLM's inference on NVIDIA A40 (we set batch size to 2). Table 7 presents the average computation time, with XGLM as the LLM. The major bottleneck of computation time lies in syntax parsing, which is due to the large size of the example database.

Reproducibility All the experiments are reproducible since all the methods are deterministic and sampling is disabled during LLM generation.

Artifact	License
spaCy	MIT
Sacremoses	MIT
retriv	MIT
XGLM	MIT
Alpaca	Apache-2.0
COMET	Apache-2.0
sacreBLEU	Apache-2.0
FLORES-101	CC-BY-SA-4.0
Europarl	Unknown
ParaCrawl	CC0
CTQ Scorer	MIT

Table 8: Licenses of scientific artifacts we use.

Scientific Artifacts We cite all the creators of scientific artifacts we use in this paper. Licenses of these scientific artifacts are shown in Table 8. Our use of these artifacts is consistent with their intended use.

References

Sweta Agrawal, Chunting Zhou, Mike Lewis, Luke Zettlemoyer, and Marjan Ghazvininejad. 2023. In-context examples selection for machine translation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8857–8873, Toronto, Canada. Association for Computational Linguistics.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online. Association for Computational Linguistics.

Elias Bassani. 2023. retriv: A Python Search Engine for the Common Man.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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. In *Advances in Neural Information Processing Systems*,

volume 33, pages 1877–1901. Curran Associates, Inc.

Laming Chen, Guoxin Zhang, and Eric Zhou. 2018. Fast greedy map inference for determinantal point process to improve recommendation diversity. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Susan Craw. 2017. Manhattan distance. *Encyclopedia of Machine Learning and Data Mining*, pages 790–791.

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. *Preprint*, arXiv:2301.00234.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Marjan Ghazvininejad, Hila Gonen, and Luke Zettlemoyer. 2023. Dictionary-based phrase-level prompting of large language models for machine translation. *Preprint*, arXiv:2302.07856.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Ping Guo, Yubing Ren, Yue Hu, Yunpeng Li, Jiarui Zhang, Xingsheng Zhang, and Heyan Huang. 2024. Teaching large language models to translate on low-resource languages with textbook prompting. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15685–15697, Torino, Italia. ELRA and ICCL.

Shivanshu Gupta, Matt Gardner, and Sameer Singh. 2023. Coverage-based example selection for in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13924–13950, Singapore. Association for Computational Linguistics.

Zhiwei He, Tian Liang, Wenxiang Jiao, Zhuosheng Zhang, Yujiu Yang, Rui Wang, Zhaopeng Tu, Shuming Shi, and Xing Wang. 2024. Exploring Human-Like Translation Strategy with Large Language Models. *Transactions of the Association for Computational Linguistics*, 12:229–246.

John L Hickman. 1980. A note on the concept of multiset. *Bulletin of the Australian Mathematical society*, 22(2):211–217.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Baijun Ji, Xiangyu Duan, Zhenyu Qiu, Tong Zhang, Junhui Li, Hao Yang, and Min Zhang. 2024. Submodular-based in-context example selection for LLMs-based machine translation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15398–15409, Torino, Italia. ELRA and ICCL.

Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. 2021. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 478–494, Online. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*, pages 79–86, Phuket, Thailand.

Aswanth Kumar, Ratish Puduppully, Raj Dabre, and Anoop Kunchukuttan. 2023. CTQScorer: Combining multiple features for in-context example selection for machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7736–7752, Singapore. Association for Computational Linguistics.

Xiaonan Li, Kai Lv, Hang Yan, Tianyang Lin, Wei Zhu, Yuan Ni, Guotong Xie, Xiaoling Wang, and Xipeng Qiu. 2023. Unified demonstration retriever for in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4644–4668, Toronto, Canada. Association for Computational Linguistics.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Veselin Stoyanov, and Xian Li. 2022. Few-shot learning with multilingual generative language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Haoyu Liu, Jianfeng Liu, Shaohan Huang, Yuefeng Zhan, Hao Sun, Weiwei Deng, Furu Wei, and Qi Zhang. 2024. se^2: Sequential example selection for in-context learning. *Preprint*, arXiv:2402.13874.

Pengyu Liu, Tinghao Feng, and Rui Liu. 2022. Quantifying syntax similarity with a polynomial representation of dependency trees. *Preprint*, arXiv:2211.07005.

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? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yasmin Moslem, Rejwanul Haque, John D. Kelleher, and Andy Way. 2023. Adaptive machine translation with large language models. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 227–237, Tampere, Finland. European Association for Machine Translation.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

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

Ratish Puduppully, Anoop Kunchukuttan, Raj Dabre, Ai Ti Aw, and Nancy Chen. 2023. DecoMT: Decomposed prompting for machine translation between related languages using large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4586–4602, Singapore. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. Unbabel's participation in the WMT20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 911–920, Online. Association for Computational Linguistics.

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.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 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. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2655–2671, Seattle, United States. Association for Computational Linguistics.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

David Vilar, Markus Freitag, Colin Cherry, Jiaming Luo, Viresh Ratnakar, and George Foster. 2023. Prompting PaLM for translation: Assessing strategies and performance. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15406–15427, Toronto, Canada. Association for Computational Linguistics.

Philip Williams and Philipp Koehn. 2014. Syntax-based statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, Doha, Qatar. Association for Computational Linguistics.

Shuangzhi Wu, M. Zhou, and Dongdong Zhang. 2017. Improved neural machine translation with source syntax. In *International Joint Conference on Artificial Intelligence*.

Zhao Yang, Yuanzhe Zhang, Dianbo Sui, Cao Liu, Jun Zhao, and Kang Liu. 2023. Representative demonstration selection for in-context learning with two-stage determinantal point process. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. 2023. Compositional exemplars for in-context learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume

11

202 of *Proceedings of Machine Learning Research*, pages 39818–39833. PMLR.

Biao Zhang, Barry Haddow, and Alexandra Birch. 2023. Prompting large language model for machine translation: A case study. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 41092–41110. PMLR.

Chen Zhang, Xiao Liu, Jiuheng Lin, and Yansong Feng. 2024. Teaching large language models an unseen language on the fly. *Preprint*, arXiv:2402.19167.

Shaolin Zhu, Menglong Cui, and Deyi Xiong. 2024. Towards robust in-context learning for machine translation with large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16619–16629, Torino, Italia. ELRA and ICCL.

A Analysis of Tree-to-Polynomial Algorithms

A.1 Original Algorithm from Liu et al. (2022)

We denote the cost of the algorithm if the tree has n nodes by $T(n)$ (then $T(1) = O(1)$), the number of nodes in the tree rooted in node m by $| m |$, the number of terms in the polynomial of node m by $\| m \|$. Note that if $| m | = n$, then

$$\sum_{i=1}^{k} | n_i | = n - 1. \tag{11}$$

For simplicity, we assume that the cost of addition of polynomial terms is the same as that of multiplication.

To get the polynomial of m^l in Equation 1, we need to compute the polynomial of each n_i (each is $T(| n_i |)$ and the sum is $T_1(n) = \sum_{i=1}^{k} T(| n_i |)$) and the multiplication of the former polynomials (which is the sum of the multiplication of all possible combinations of terms from the child nodes and each combination requires multiplying k terms together plus an addition thus the overall cost should be $T_2(n) = O((1+(k-1)) \cdot \prod_{i=1}^{k} \| n_i \|)$ [4]. Then the overall cost of computing Equation 1 is

$$T(n) = T_1(n) + T_2(n)$$
$$= \sum_{i=1}^{k} T(| n_i |) + O(k \cdot \prod_{i=1}^{k} \| n_i \|). \tag{12}$$

---

[4]Here the additions include the addition of the whole product of former polynomials and y_l.
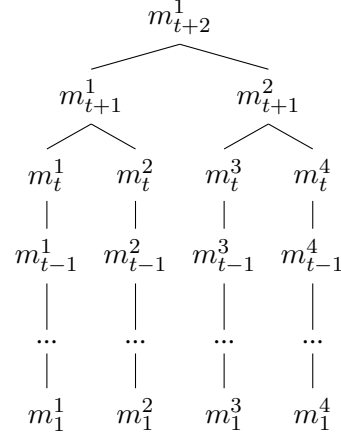

Figure 2: An example tree with $t + 2$ layers and $4t + 3$ nodes. m_i^j denotes the j-th node on the i-th layer.

Consider a tree as shown in Figure 2 with $t + 2$ layers and $4t + 3$ nodes, where m_i^j denotes the j-th node on the i-th layer. The cost of computing the polynomial of m_t^j should be $O(t)$ since it is just to add t single-variable terms together according to Equation 1. Then, the cost of computing the polynomial of m_{t+1}^i should be $O(2t + 2t^2)$ according to Equation 12, which can be further simplified to $O(t^2)$. Finally, the cost of computing the polynomial of m_{t+2}^1, which is also the polynomial representing the whole tree, should be $O(2t^2 + 2(t^2)^2)$ according to Equation 12, which can be further simplified to $O(t^4)$. Thus in this tree, the cost is:

$$\hat{T}(4t + 3) = O(t^4). \tag{13}$$

Let $s = 4t + 3$, we simplify Equation 13 and have:

$$\hat{T}(s) = O(s^4). \tag{14}$$

Therefore, we prove that the original tree-to-polynomial algorithm can be of quartic time complexity in some cases.

In fact, given any constant $p = 2^q$ where q is a positive integer, we can construct a tree in the way as shown in Figure 2 with $t + q$ layers and $pt + p - 1$ nodes. Let m_{t+q}^1 denote the root node. For each i between 1 and q and each j between 1 and 2^{q-i}, m_{t+i}^j has two child nodes m_{t+i-1}^{2j-1} and m_{t+i-1}^{2j}. For each i between 2 and t and each j between 1 and 2^q, m_i^j has only one child node m_{i-1}^j. Finally, for each j between 1 and 2^q, m_1^j is the leaf node. In this way, the cost of computing the polynomial of m_t^j is $O(t)$ as discussed above. That of m_{t+1}^j, m_{t+2}^j, ..., m_{t+q}^j should be $O(t^2)$, $O(t^4)$, ..., $O(t^{2^q})$ recursively, the last one with q times of recursion

being $\hat{T}(pt+p-1)$ indeed. Let $s = pt+p-1$, we again ignore the constant factors and insignificant terms and then have:

$$\hat{T}(s) = O(s^{2^q}) = O(s^p). \tag{15}$$

Thus we prove that the complexity of the original tree-to-polynomial algorithm can be polynomial of arbitrarily large degree $p = 2^q$ in some cases. So when dealing with very large dependency trees of long sentences, the original algorithm can be quite time-consuming and thus impractical for application in MT where there can be millions of data to be processed.

However, we have not proven the exact lower bound of the cost or the average cost according to Equation 12, which we leave for future work.

A.2 Our Simplified Algorithm

We use the same symbols as in Section A.1. Given $\mid m \mid = n$, Equation 11 still holds in this section. Moreover, in our simplified algorithm, the number of terms in a polynomial equals to the number of nodes in the tree rooted in the corresponding node:

$$\parallel m \parallel = \mid m \mid, \tag{16}$$

and

$$\parallel n_i \parallel = \mid n_i \mid. \tag{17}$$

To get the polynomial of m^l in Equation 2, we need to compute the polynomial of each n_i (each is $T(\mid n_i \mid)$), the sum of 1 and all the former polynomials (which is $O(\sum_{i=1}^{k} \parallel n_i \parallel)$), the multiplication of x_l (which can be seen as multiply x_l with all the terms in the former polynomials plus 1 and thus should be $O(1 + \sum_{i=1}^{k} \parallel n_i \parallel)$ and can be further simplified to $O(\sum_{i=1}^{k} \parallel n_i \parallel)$). Then the overall cost of computing Equation 2 is

$$T(n) = \sum_{i=1}^{k} T(\mid n_i \mid) + 2 \cdot O(\sum_{i=1}^{k} \parallel n_i \parallel). \tag{18}$$

We then apply Equation 11 and 17 and ignore the constant factors to get

$$T(n) = \sum_{i=1}^{k} T(\mid n_i \mid) + O(n). \tag{19}$$

Then

$$T(n) - \sum_{i=1}^{k} T(\mid n_i \mid) = O(n). \tag{20}$$

Analogously,

$$\forall 1 \leq i \leq k, T(\mid n_i \mid) - \sum_{j=1}^{k_i} T(\mid n_{i_j} \mid) = O(\mid n_i \mid), \tag{21}$$

where n_i has k_i child nodes denoted by n_{i_j}. Thus

$$\sum_{i=1}^{k} T(\mid n_i \mid) - \sum_{i=1}^{k} \sum_{j=1}^{k_j} T(\mid n_{i_j} \mid) = \sum_{i=1}^{k} O(\mid n_i \mid)$$
$$= O(n-1). \tag{22}$$

With the recursive boundary

$$T(1) - 0 = O(1), \tag{23}$$

we can continue the process recursively (in fact, each level of recursion corresponds to a layer in the tree) until each node has appeared on left-hand side and add together Equation 20, 22 and so on to get

$$T(n) = O(n) + O(n-1) + O(n-1-k) + ...$$
$$\leq O(n^2). \tag{24}$$

Thus we prove that the complexity of our simplified tree-to-polynomial algorithm is no more than quadratic time.

B BLEU Results

The BLEU scores of our main results are shown in Table 9.

C The spaCy Models Used for Parsing

The spaCy models and their corresponding versions we use for dependency parsing are listed in Table 10.

D Prompt Template

For XGLM, we use the same prompt template as used in Kumar et al. (2023):

```
[source] sentence: [X_1]
[target] sentence: [Y_1]
###
...
[source] sentence: [X_k]
[target] sentence: [Y_k]
###
[source] sentence: [X]
[target] sentence:
```

13

System	Into EN			Out of EN			Avg.
	DE	FR	RU	DE	FR	RU	
XGLM							
Learning-free							
Random	31.31	32.68	**24.85**	19.63	28.79	17.57	25.81
BM25	31.06	**33.34**	24.47	20.16	**29.79**	18.18	26.17
R-BM25	31.16	32.99	24.71	20.00	29.17	17.93	25.99
Fuzzy	**31.95**	33.08	24.42	20.29	29.77	18.01	**26.25**
SCOI (*ours*)	31.51	32.88	**24.85**	**20.45**	29.39	**18.25**	26.22
Learning-based							
CTQ Scorer	32.02	32.35	25.29	20.94	30.59	18.53	26.62
Alpaca							
Learning-free							
Random	33.50	**36.24**	26.48	20.08	29.05	15.82	26.86
BM25	33.16	35.11	26.58	20.23	**29.76**	15.99	26.81
R-BM25	33.47	35.42	26.23	20.46	29.64	16.27	26.92
Fuzzy	33.51	35.51	26.02	20.26	29.58	15.87	26.79
SCOI (*ours*)	**33.93**	35.44	**26.69**	**20.70**	29.61	**16.53**	**27.15**
Learning-based							
CTQ Scorer	33.75	35.83	26.56	20.99	30.23	16.26	27.27

Table 9: BLEU scores of 4-shot ICL performance of SCOI and other methods for translation on all 6 directions of 2 language models. All the methods except CTQ Scorer are learning-free, which do not require task, language or LLM-specific training. "**Avg.**" refers to the average score across all 6 directions. The highest scores among learning-free methods are in **bold** text.

Language	spaCy Model	Version
DE	de_core_news_sm	3.7.0
EN	en_core_web_sm	3.7.1
FR	fr_core_news_sm	3.7.0
RU	ru_core_news_sm	3.7.0

Table 10: The spaCy models and their versions of different languages used for dependency parsing.

In the template, [source] and [target] refer to the names of the source and target languages in English (e.g., German, French, etc.). The ### symbol is used as an example delimiter and a marker for answer extraction in post-processing.

With the same symbols above, for Alpaca, we use the same prompt template as used in He et al. (2024):

```
Instruction: Translate the following
[source] text into [source].

[source]: [X_1]
[target]: [Y_1]
...
[source]: [X_k]
[target]: [Y_k]
[source]: [X]
```

[target]:

E Implementation Details

Noting that our test data and example databases are the same as those used in Kumar et al. (2023), we directly use the examples selected by BM25, R-BM25 and CTQ Scorer from Kumar et al. (2023) [5].

Following Kumar et al. (2023), we remove instances in the example database with more than 120 tokens in order to avoid overlong context.

F Effect of Parser

Direction	Δ	Parser	LAS
DE-EN	+1.46	de_core_news_sm	0.90
FR-EN	-0.10	fr_core_news_sm	0.83
RU-EN	+1.60	ru_core_news_sm	0.95
Out of EN (Avg.)	+0.80	en_core_web_sm	0.90

Table 11: Performance gains ("Δ") of SCOI over BM25 using XGLM and capabilities of corresponding parsers on different translation directions. "LAS" refers to the labeled attachment score of a parser.

In order to better understand the relation between the performance of SCOI and the capability of the parser, we compare the labeled attachment scores

---

[5]https://github.com/AI4Bharat/CTQScorer

System	Into EN			Out of EN			Avg.
	DE	FR	RU	DE	FR	RU	
Learning-free							
Random	77.52	81.78	**67.02**	69.04	84.12	71.26	75.12
BM25	**77.54**	81.60	66.17	68.93	84.06	71.73	75.01
R-BM25	77.24	81.54	66.45	**69.25**	84.08	71.46	75.00
Fuzzy	77.36	**81.89**	66.52	68.83	**84.33**	**72.49**	**75.24**
SCOI (*ours*)	77.17	**81.89**	66.38	69.07	84.31	72.13	75.16
Learning-based							
CTQScorer	77.40	81.99	66.77	69.33	83.78	73.06	75.39

Table 12: Results on GPT-3.5.

(LAS) of different parsers used in our experiments reported on the official website of spaCy [6]. Table 11 shows performance gains of SCOI over the BM25 baseline using XGLM and capabilities of corresponding parsers. The results show that a better parser leads to better performance of SCOI and incidate that SCOI is highly dependent on parsers.

G Results on GPT-3.5

We call OpenAI's API [7] of `gpt-3.5-turbo-0125` to evaluate different in-context example selection methods on GPT-3.5. Results are presented in Table 12.

It seems the difference between in-context example selection methods is not so significant as that on smaller LLMs. This might be because that the capability of GPT-3.5 has been strong enough so that in-context examples bring limited help. For such large-scaled models, design and organization of prompt and use of additional information or knowledge might be more crucial in improving performance of ICL.

H Polynomial Distance

Note that we can write polynomial terms as term vectors as shown in Equation 5. In this way, a polynomial P can be written as a set of term vectors $\mathcal{V}_P$. Then, following Liu et al. (2022), we compute the distance between two polynomials (P and Q) as:

$$d(P, Q) = \frac{\sum\limits_{s \in \mathcal{V}_P} \min\limits_{t \in \mathcal{V}_Q} \| s - t \|_1 + \sum\limits_{t \in \mathcal{V}_Q} \min\limits_{s \in \mathcal{V}_P} \| s - t \|_1}{| \mathcal{V}_P | + | \mathcal{V}_Q |},$$
(25)

where $\| s - t \|_1$ is the Manhattan distance (Craw, 2017) between term vector s and t.

I Details of DPPs

We set the λ in Equation 9 to 0.5 to balance syntactic relevance and lexical diversity. As mentioned in Section 5.3, the word vectors $\mathbf{W}_{N \times T}$, used to compute lexical diversity, where N is the number of documents (candidate examples) and T is the number of terms (words) in each test input, are constructed as follows:

$$\mathbf{W}_{i,j} = \mathrm{idf}_j \times \left(\frac{\mathrm{tf}_{i,j} \times (k_1 + 1)}{\mathrm{tf}_{i,j} + k_1 \times (1 - b + b \times l_i)} \right),$$
(26)

where i and j refer to the i-th candidate example and the j-th term in a test input, respectively. Here, idf_j is the inverse document frequency of the j-th term across all candidate examples, $\mathrm{tf}_{i,j}$ is the term frequency of the j-th term in the i-th candidate example, and l_i is the length of the i-th candidate example. The parameters k_1 and b are hyperparameters.

J Effect of Measure of Coverage

Coverage	Into EN			Out of EN			Avg.
	DE	FR	RU	DE	FR	RU	
Cosine Similarity	64.35	**71.54**	53.89	**45.41**	**55.36**	46.06	56.10
Normalized Distance	**64.67**	71.26	**54.08**	44.87	55.31	**46.47**	**56.11**

Table 13: COMET scores of 4-shot ICL performance on XGLM of different measures of coverage.

As described in Section 3.2, we compute the coverage of polynomial terms $\mathsf{c}(s, t)$ in Equation 4 by Equation 6 and 7, which is the normalized Manhattan distance between two term vectors. For comparison, we also explore cosine similarity as the measure of coverage:

$$\mathsf{c}(s, t) = \frac{v_s \cdot v_t}{\| v_s \| \| v_t \|},$$
(27)

where v_s and v_t are the vectors described in Equation 5 representing terms s and t respectively. Thus,

15

	DE	EN
Input & Gold	Nach einer Woche voller Verluste in der Zwischenwahl erzählte Bush dem Publikum von der Ausweitung des Handels in Asien.	After a week of losses in the midterm election, Bush told an audience about the expansion of trade in Asia.
Example-1	Ich stimme mit dem Standpunkt der Berichterstatterin überein und bin mit den eingeführten Veränderungen, wie der Ausweitung der Mindestdauer des Mutterschaftsurlaubs von 14 auf 20 Wochen, dem Grundsatz einer Bezahlung in voller Höhe des bisherigen Einkommens, der Einführung von Gesundheitsschutzbestimmungen am Arbeitsplatz und dem Verbot der Kündigung, einverstanden.	I agree with the position of the rapporteur and with the changes introduced, such as the extension of the minimum period for maternity leave from 14 to 20 weeks, the principle of pay equivalent to complete earnings, the establishment of health and safety requirements in the workplace, and the prohibition of dismissal.
Example-2	Deshalb geht meiner Ansicht nach der Verlust von Sprachen mit dem Verlust von Lebensweisen einher.	I think, therefore, that if we lose languages we lose forms of life.
Example-3	Herr Minister, diese Woche wird von dem erklärten Willen des Europäischen Parlaments geprägt sein, gegen den Verlust der biologischen Vielfalt anzukämpfen.	Minister, this week will have been marked by the desire shown by the European Parliament to fight against the loss of biodiversity.
Example-4	Nach dem, was mir erzählt wurde, nicht gut.	From what I was told I suspect they were not good.
Prediction	-	After a week of losses in the mid-election campaign, President Bush told his audience that trade in Asia had been expanded.

Table 14: An end-to-end "DE-EN" translation example of BM25, with the same test input in Table 6.

$c(s, t)$ is measured by the cosine similarity between v_s and v_t.

Experimental results are shown in Table 13. The difference of performance between the two measures is not significant and thus we infer that the measure of coverage has little effect on the performance of SCOI.

K The Example of BM25

An end-to-end German-to-English translation example of BM25 is shown in Table 14, the test input is the same as that of SCOI discussed in Section 5.5.

BM25 mainly focuses on lexical similarity and does not take coverage into consideration. For example, the word "Publikum" is not covered by BM25 since it is based on the Top-k mode while SCOI does cover it. Moreover, it does not emphasize the similarity in syntax. Even though some examples contain similar syntactic structure (e.g., the second example is just the first example selected by our method), BM25 fails to put these examples in the front to allow LLMs pay more attention to those more helpful examples.