SSP: Self-Supervised Prompting for Cross-Lingual Transfer to Low-Resource Languages using Large Language Models

Anonymous ACL submission

Abstract

Recently, very large language models (LLMs) have shown exceptional performance on several English language NLP tasks with just in-context learning (ICL), but their utility in other languages is still underexplored. We investigate their effectiveness for NLP tasks in low-resource languages (LRLs), especially in the setting of zero-shot cross-lingual transfer (0-CLT), where task-specific training data for one or more related medium-resource languages (MRLs) is available. We introduce Self-Supervised Prompting (SSP), a novel ICL approach for the 0-CLT setting.

SSP is based on the key observation that LLMs output more accurate labels if in-context exemplars are from the target language (even if their labels are slightly noisy). To operationalize this, since target language training data is not available in 0-CLT, SSP operates in two stages. In Stage I, using source MRL training data, target language's test data is noisily labeled. In Stage II, these noisy test data points are used as exemplars in ICL for further improved labeling. Additionally, our implementation of SSP uses a novel Integer Linear Programming (ILP)-based exemplar selection that balances similarity, prediction confidence (when available) and label coverage. Experiments on three tasks and twelve LRLs (from three regions) demonstrate that SSP strongly outperforms fine-tuned and other prompting-based baselines.

1 Introduction

Very large language models (LLMs) such as GPT-3.5-Turbo & GPT-4 (Ouyang et al., 2022; Achiam et al., 2023) show exceptional performance on a variety of NLP and reasoning tasks via *In-Context Learning* (ICL) (Brown et al., 2020; Chowdhery et al., 2022). ICL feeds a task-specific instruction along with a few exemplars, appended with the test input, to the LLM. As LLMs can be highly sensitive to exemplars (Zhao et al., 2021), exemplar retrieval is crucial for ICL.

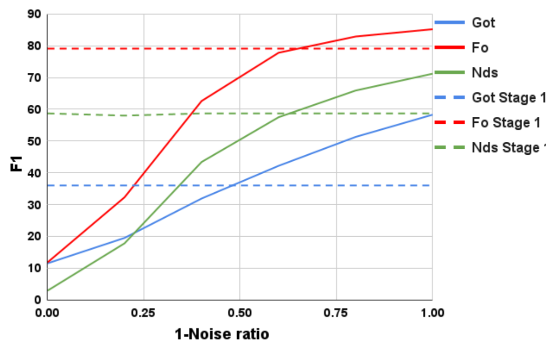

Figure 1: Llama2 70B, prompted with target LRL exemplars, along with artificially injected label noise (x-axis). Dashed lines represent performances when prompted with source MRL exemplars.

While LLMs have shown excellent performance on English tasks, their utility on other languages is relatively underexplored. In this work, we study *zero-shot cross-lingual transfer* (0-CLT) to low-resource languages (LRLs) – a setting where labeled task data from one or more related medium-resource languages (MRLs) is available, but no labeled training data exists for the target LRL.

Cross-lingual transfer has been addressed through standard fine-tuning (Muller et al., 2021; Alabi et al., 2022), and language adapters (Pfeiffer et al., 2020; Üstün et al., 2020; Rathore et al., 2023), but there is limited work on cross-lingual ICL. There are two exceptions (Ahuja et al., 2023; Asai et al., 2023), where ICL is employed with exemplars from a source language, but they use uniformly random sampling for exemplar selection, resulting in performance inferior to cross-lingually fine-tuned models, such as mBERT and XLM-R (Devlin et al., 2019; Conneau et al., 2020).

In our preliminary experiments, we prompt the Llama2-70B model with exemplars from source MRLs, and compare it's performance with the same LLM prompted with exemplars from the target

LRL. We vary the label noise on the target exemplars. Unsurprisingly, LLMs show better performance with less label noise. More interestingly, we find that a reasonably-sized noise region exists (see Figure 1), such that if the exemplar noise is within that range, then the overall performance is higher than prompting with source language data.

Armed with this observation, we present Self-Supervised Prompting (SSP) – a novel ICL framework for 0-CLT to LRLs. Since the target LRL training data is not available in 0-CLT, SSP operates in two stages. In Stage I, SSP labels all test instances of LRL using training data from MRL. This may be done by LLM prompting (as in the experiment above), or using any other existing approaches for 0-CLT, such as by fine-tuning or adapters. Once (noisy) labels on target LRL are obtained, in Stage II, SSP uses ICL using these noisy test data points (except itself) as exemplars for further performance improvement. Additionally, to select the best exemplars, we develop a novel Integer Linear Programming (ILP) based selection approach, which balances the various objectives of (1) similarity of exemplar with test sentence, (2) high confidence in label predictions, and (3) coverage of the various labels for better task understanding. Figure 2 gives an overview of our proposed pipeline.

We perform experiments on sequence labeling tasks (POS and NER), and natural language inference (NLI) – a text classification task. Our datasets encompass twelve low-resource languages from typologically diverse language families and three regions: African, Germanic and American. Our experiments show consistent and substantial improvements over existing fine-tuning as well as simpler ICL-based approaches. We will make both our codebase and prompts publicly accessible.

Our contributions are summarized as follows:

1. We investigate ICL strategies for the task of zero-shot cross-lingual transfer to low-resource languages, utilizing the labeled data from related languages.
2. We propose SSP, a two-stage self-adaptive prompting paradigm for this task, where the first stage may be done by an LLM or other cross-lingual transfer models.
3. We introduce an exemplar selection approach that utilizes an ILP. The ILP incorporates similarity to test input along with confidence of prediction (when available), and enforces label coverage constraints for better selection.
4. Experiments on 3 tasks and 11 languages show that SSP outperforms existing fine-tuning, adapter and LLM-based SoTA models.

2 Related Work

An ICL prompt consists of (1) task description: to facilitate the understanding of task, (2) labeled input-output pairs: Written sequentially in order of their relevance to input query, and (3) input itself.

Cross-lingual ICL: In general, cross-lingual ICL has not been systematically explored in literature. In existing works, prompting is primarily done in a high-resource language, typically English. This is called *cross-lingual (CL) prompting*. This differs from *in-language (IL) prompting*, where examples are retrieved from the candidate pool of the target language itself. This assumes the availability of labeled data for target LRL, which is not true in our zero-shot setting. In response, we develop novel techniques making use of both CL prompting and IL prompting, while not utilizing the gold labels during IL prompting stage.

Most existing cross-lingual ICL methods use uniformly random input-output pairs for exemplar selection (Zhang et al., 2021; Winata et al., 2021; Ahuja et al., 2023; Asai et al., 2023). Recent approaches (Agrawal et al., 2022; Tanwar et al., 2023) address this gap by utilizing *semantic similarity* for cross-lingual retrieval from a high-resource language's labeled data, given the target LRL's instance as query. This is facilitated by embedding-based multilingual retrievers such as multilingual sentence-transformers (Reimers and Gurevych, 2020). More recently, OpenAI-based embeddings such as Ada-002[1] have been used effectively for cross-lingual retrieval (Nambi et al., 2023). We extend this line of work by also incorporating label confidence and label coverage in exemplar selection.

Self-Adaptive Prompting: Wan et al. (2023) proposed *Universal Self-Adaptive* (USP) framework, which has been explored only for monolingual (English) setting. USP uses an external *unlabeled* dataset of instances and labels them using LLM in Stage I. It then samples multiple Chain-of-thought (CoT) paths to estimate the logits using the same LLM, and then utilizes the entropy of logits for exemplar selection for Stage 2. Our work has similarities to USP in that both methods are two-stage

[1]https://platform.openai.com/docs/guides/embeddings/embedding-models

prompting approaches. USP is different from SSP in that the former is much more expensive, since it requires multiple LLM runs to estimate logits. USP also does not use any exemplars (and only uses task description), which are quite important for better performance. Finally, USP has only been applied for English tasks, and has not been explored for cross-lingual tasks.

Fine-tuning approaches for Cross-lingual Transfer: Most approaches rely on fine-tuning a Pre-trained LM (PLM) such as BERT or XLM-R on one or more source languages ((Muller et al., 2021; Alabi et al., 2022)) and deploying on an unseen target language. Recently, Language-Adapter based approaches have been found more effective (Üstün et al., 2020) for cross-lingual transfer settings. For sequence labeling tasks (NER and POS tagging), ZGUL (Rathore et al., 2023) is a recent SOTA method that leverages ensembling Language Adapters from multiple MRLs to label each word in a target language. We leverage this in our proposed SSP pipeline.

3 Self-Supervised Prompting

We define the setting of zero-shot cross-lingual transfer (0-CLT) as follows. We are given source training data for a specific task: $D = \{(x_i, lg_i, y_i)\}$, where x_i is the input text in language lg_i, and the output is y_i. We are additionally given a set of unlabeled test data points $T = \{q_j\}$ from a target language lg_t. Our goal is to train a model/create a protocol, using D, T and a large pre-trained LLM, that outputs good predictions on T for the task, assuming that lg_t is a low-resource language, due to which its training data is not available, and that languages lg_i are related to lg_t.

Our solution approach, Self-Supervised Prompting (SSP), comprises two key stages as follows. In Stage I, it proposes a noisy labeling for all data points in T using source data D. This may be done in different ways, as described next. In Stage II, it uses the LLM and noisy labeling on T from Stage I as exemplars to improve the labelings. Furthermore, SSP uses a novel integer-linear programming based exemplar selection. We now describe each component of our system.

3.1 Stage I: initial labeling using source data

To create a first labeling for all test points, SSP can use any existing approaches for 0-CLT, such as fine-tuning a multilingual language model for the task, or use of language adapters or using our LLM with in-context exemplars from source language. In our experiments, we experiment with adapters and ICL, which we briefly describe next.

Cross-Lingual ICL: In the method, we use ICL over LLM for obtaining Stage I labelings. First, we retrieve a set of top-K exemplars from D using each test instance q_j as query. This selection is based on cosine similarity between their *Ada-002* embeddings. The selected exemplars are arranged in descending order of similarity scores, and included in the prompt between the task description (TD) and the input test instance. This approach has two drawbacks. First, since the LLM will typically be a large expensive model – this will require an LLM call per test data point in Stage I. Second, generally, these LLMs do not expose their logits, hence, we will not have access to prediction confidences from Stage I labelings.

Training smaller model(s) using D: Another possibility is to fine-tune a smaller multilingual LM, such as mBERT or mDeBerta-v3 (He et al., 2021) on D for NLI task. For sequence labeling, we can use ZGUL (Rathore et al., 2023), which trains source language adapters using D, and uses inference-time fusion of source adapters for labeling test data points. These approaches can provide Stage I labelings for T along with prediction confidences, without making any expensive LLM calls.

3.2 Stage II: in-language ICL using ILP-based exemplar selection

After Stage I predictions for target instances T are obtained, SSP prompts the LLM to label each test data point $q \in T$, but uses in-context exemplars in target language using Stage I labelings. For exemplar selection, SSP implements a novel integer linear program (ILP) that balances *semantic similarity, prediction confidence* (when available) and *label coverage*.

Our primary objective is to maximize the aggregated semantic similarity of the selected exemplars, which is obtained using cosine similarity score between their OpenAI Ada-v2 embeddings. In addition, we impose two constraints:

- **Label Coverage**: The ILP tries to ensure the coverage of all labels for the given task in the selected exemplars – this has been found effective for ICL (Min et al., 2022).

- **Confidence**: In case Stage I predictions are

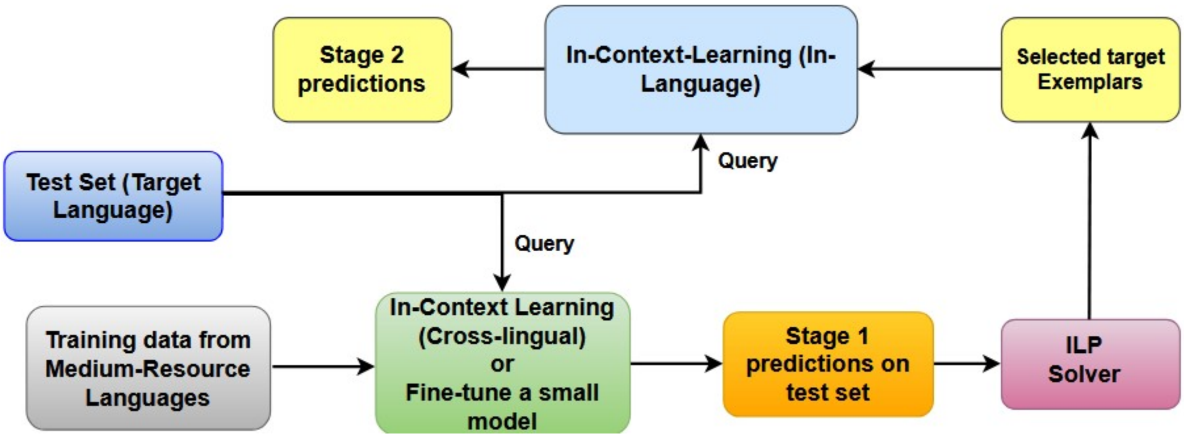

Figure 2: SSP Paradigm for Cross-Lingual Transfer to target low-resource language

made by a model whose logits are accessible (unlike the OpenAI LLMs), the ILP prefers selection of more confident exemplars. Our hypothesis is that confident predictions are also accurate (assuming the model is well-calibrated), and previous work has shown that performance of LLMs can be sensitive to correctness of exemplars (Wei et al., 2023)

SSP formulates these three factors into an ILP as follows. For a dataset D with n examples indexed from $\mathcal{I} = \{1 \ldots n\}$, given a test data point q_j, let z_i be a binary variable denoting whether i^{th} test instance q_i is selected as an exemplar. We use a semantic similarity function $\text{sim}(q_i, q_j)$ to get the similarity between two examples. K is the number of exemplars to be selected. Since q_j cannot be an exemplar for itself, we select exemplars from $\mathcal{I} \setminus \{j\}$ only.

Let the set of all labels in the task be $\mathcal{L}$, and the multiset of all labels predicted (using argmax) for example q_i be L_i. The Stage I prediction confidence for label l in q_i is denoted as $\hat{y}_l^i$. This confidence is computed as average of probability scores across all predictions of label l in i^{th} sentence (details in Appendix A). The ILP uses a threshold τ_l for prediction confidence for a label l. Intuitively, the ILP maximizes the semantic similarity of K chosen exemplars, subject to each label l being present at least once in the exemplars, and average prediction confidence of each data point for each label being greater than τ_l.[2]

[2]Although we express constraints (3) and (4) as a hard constraint, they are implemented as soft constraints (added in the primary objective) following standard practices of approximate solvers such as Gurobi

Formally, the ILP is formulated as

$$\max \sum_{i \in \mathcal{I} \setminus \{j\}} z_i \cdot \text{sim}(q_i, q_j) \quad (1)$$

$$\text{such that} \sum_{i \in \mathcal{I} \setminus \{j\}} z_i = K \quad (2)$$

$$z_i \cdot (\hat{y}_l^i - \tau_l) \geq 0 \;\forall\, i \in \mathcal{I} \setminus \{j\}, \forall\, l \in L_i \quad (3)$$

$$\sum_{i \in \mathcal{I} \setminus \{j\}} z_i \cdot \text{count}(L_i, l) \geq 1 \;\forall\, l \in \mathcal{L} \quad (4)$$

Here $\text{count}(L_i, l)$ denotes the number of occurences of l in L_i. In our experiments, we set $K = 8$, and τ_l = 80^{th} percentile threshold of the set $\{\hat{y}_l^i\}_{i=1}^n$ for a particular label l. The idea is to have label-specific threshold since the fine-tuned model may not have same calibration for all labels.

Since logits are not accessible for OpenAI LLMs GPT-3.5 and GPT-4x, in case Stage I labeling is done by either of these models using ICL, we skip the confidence thresholding constraint of ILP. This means that for this variant of SSP, the selection is made based on only similarity and label coverage.

4 Experiments

Our main experiments assess SSP performance compared to existing state-of-the-art models for 0-CLT. We also wish to compare various SSP variants, and estimate the value of the ILP-based exemplar selection.

4.1 Tasks and Datasets

We experiment on three tasks – POS tagging, NER and Natural Language Inference (NLI). We use the Universal Dependency dataset (Nivre et al., 2020) for POS tagging over Germanic languages,

Model	Hau	Ibo	Kin	Lug	Luo	Avg.	Fo	Got	Gsw	Avg
Full Fine-Tuning (FFT)	49.9	54.9	55.4	56.3	40.2	51.3	77.6	17.8	62	52.5
CPG (Üstün et al., 2020)	48.6	50.4	52.6	54.3	38.6	48.9	77.3	16.9	63.9	52.7
ZGUL	52.2	56	53.7	54.5	44.4	52.2	77.2	21.1	65	54.4
ICL-Llama-2-70b	64.3	61.2	59.2	60.1	47.3	58.4	79.1	36.0	71.8	62.3
ICL-GPT-3.5-turbo	54.5	69.2	57.8	63.7	46.4	58.3	81.2	37.9	72.2	63.8
ICL-GPT-4x	64.7	80.8	64.6	71.0	53.3	66.9	81.3	66.5	82.3	76.7
SSP(ICL)-llama-2-70b	57.6	62.6	56.0	57.6	43.1	55.4	78.5	37.9	73.5	63.3
SSP(ICL)-GPT-3.5-turbo	62.8	68.4	64.0	63.8	47.6	61.3	82.4	63.2	79.4	75.0
SSP(ICL)-GPT-4x	67.2	79.6	63.3	74.1	54.4	67.7	81.8	**73.7**	85.4	**80.3**
SSP(ZGUL)-Llama-2-70b	68.4	58	56.1	54.7	42.3	55.9	79.9	39.9	72.9	64.2
SSP(ZGUL)-GPT-3.5	61.1	68.9	62.1	67.1	51.4	62.1	**82.8**	67.5	77	75.8
SSP(ZGUL)-GPT-4x	**72.5**	79.8	**71.4**	**77.4**	**55.1**	**71.2**	82.2	71.5	**85.6**	79.8
w/o Conf. thresholding	71.3	**81.9**	69.2	74.6	52.7	69.9	82.8	57	81.4	73.7
w/o Label Coverage	71.1	79.8	**71.4**	**77.4**	**55.1**	71	82.2	71.6	**85.6**	79.8
w/o both (sim-based)	70.3	81.8	68	74.8	51.9	69.4	82.4	55.8	82.3	73.5
w/o ILP (Random)	64.1	77.6	61.5	66.1	46.6	63.2	80.6	54.8	80.9	72.1
Skyline (GPT-4x)	*75.5*	*85.9*	*70.7*	*73.6*	*67.2*	*74.6*	*93.5*	*80.7*	*89.9*	*88*

Table 1: Micro-F1 scores for African NER (left) and Germanic POS (right) (Statistical significance of bold numbers: McNemar p-value = 0.008 and 0.0004, respectively)

Family	Source languages	Source size
Germanic	{En,Is,De}	30000
African	{En,Am,Sw,Wo}	19788
American	{En,Es}	19998

Table 2: Size (No. of sentences) of Combined Source language datasets (En - English, Is - Icelandic, De - German, Am - Amharic, Sw - Swahili, Wo - Woloff, Es - Spanish)

Family	Test languages	Labels
Germanic	{Fo, Got, Gsw}	2370
African	{Hau,Ibo,Kin,Lug,Luo}	1100
American	{Aym,Gn,Nah}	501

Table 3: Size (No. of labels) of Target language datasets, *per language*, on average. (Fo - Faroese, Got - Gothic, Gsw - Swiss German, Hau - Hausa, Ibo - Igbo, Kin - Kinyarwanda, Lug - Luganda, Luo - Luo, Aym - Aymara, Gn - Guarani, Nah - Nahuatl)

MasakhaNER (Adelani et al., 2021) for African NER, and AmericasNLI (Ebrahimi et al., 2022) for NLI task on the indigenous languages of Americas. Overall, we use twelve low-resource test languages as target (e.g., Kinyarwanda, Faroese, and Aymara), and 2-4 source languages per dataset (e.g., Icelandic, Spanish and Swahili; always including English). Further details are in Appendix C. Tables 2 and 3 show the languages and number of examples in the source and target datasets respectively.

Recent studies have shown sensitivity of the output to the template/format of input-output pairs written in the prompt (Sclar et al., 2023; Voronov et al., 2024). We follow the best template given in Sclar et al. (2023) for NLI, while for sequence labeling, we explore various templates on our own and report our results on the best one. We refer to Appendix B for details and the exact templates used for each of our tasks.

For obtaining test set, we randomly sample 100 test samples for each target language for NER and POS tasks. We justify this as each sentence has multiple labels, bringing the total no. of instances to be labeled per language to 2370 and 1100 for POS and NER respectively. For the NLI task, we sample 501 test samples (167 for each class: 'entailment', 'contradiction' and 'neutral'). We report statistical significance (in table captions) to justify our evaluation.

We also perform a careful contamination study, following (Ahuja et al., 2022), by asking LLMs to fill dataset card, complete sentence (and labels), given partial sentence, and generate next few instances of the dataset. As further detailed in Appendix F, we do not observe any evidence of contamination of these languages' test splits in the OpenAI LLMs, suggesting that OpenAI LLMs have likely not seen these test datasets during their training.

4.2 Comparison Models

Baselines: We compare our SSP approach with the SoTA fine tuning models, as well as LLM-based ICL methods using naive random exemplar selection. In particular, we fine-tune ZGUL – mBERT Language Adapter-based SoTA zero-shot baseline for NER and POS tagging, and mDeBERTa fine-

Model	Aym	Gn	Nah	Avg.
mDeBerta^{100} (Laurer et al., 2022)	34.9	43.9	48.9	42.6
mDeBerta^{CL}	33.9	47	46.9	42.6
ICL-GPT-3.5-turbo	38.2	41.7	35.3	38.4
ICL-GPT-4x	32.8	55.8	42.2	43.6
SSP(ICL)-GPT-3.5-turbo	38.4	38.8	43.2	40.1
SSP(ICL)-GPT-4x	37.5	58.5	51.8	49.3
SSP(ZGUL)-GPT-3.5	**43.1**	46	46.8	45.3
SSP(ZGUL)-GPT-4x	36	**61.3**	**59.2**	**52.2**
w/o Conf. thresholding	42.9	60.1	50.3	51.1
w/o Label Coverage	37	58.2	57.4	50.9
w/o both (sim-based)	34.3	59.7	57.1	50.4
w/o ILP (Random)	33.4	53.8	53.4	46.9
Skyline (GPT-4x)	*55.6*	*49.2*	*60*	*54.9*

Table 4: Micro-F1 scores for Americas NLI (Statistical significance of bold number: McNemar p-value = 0.054)

tuned for NLI. We additionally utilize the public model mDeBERTa-v3-base-xnli-multilingual-nli-2mil7 (Laurer et al., 2022) for NLI evaluation. We term our own fine-tuned model as mDeBERTa^{FT} and the public model as mDeBERTa^{100}, as it was trained on 100 languages (while not covering any of our target languages). For POS and NER, we also add full parameter fine-tuning and Conditional Parameter Generation (CPG (Üstün et al., 2020)) baselines, fine-tuned using the same underlying LM (i.e. mBERT) as ZGUL.

SSP Variants: We implement SSP with a series of top-of-the-line LLMs – GPT-3.5-turbo (Ouyang et al., 2022), GPT-4x (GPT-4/GPT-4-Turbo) (Achiam et al., 2023), and LLaMa-2-70b (Touvron et al., 2023). If Stage I uses ICL, then the same LLM is used for both stages I and II. Alternatively, ZGUL and mDeberta based methods are also used in Stage I of SSP.

To understand the value of the ILP, we perform three ablations on exemplar selection strategy – (a) without confidence thresholding (for fine-tuned LM), (b) without label coverage and (c) without both, i.e. pure similarity-based. The ablations are conducted with the best performing underlying LLM i.e. GPT-4x.

Skyline: To understand the current performance gap due to lack of target language training data, we also implement a skyline utilizing the available data for target languages and perform *few-shot in-language similarity-based* exemplar selection (using Ada-v2 embeddings) for *in-language* ICL to the LLM.

5 Results and Analysis

We present the results for all tasks in Tables 1, and 4. ICL-X represents ICL over an LLM X with source language exemplars. SSP($model$)-X represents the use of $model$ for Stage I followed by LLM X for Stage II. In case ICL is used in Stage I, then same LLM X is used in both stages.

Analyzing the results, we first observe that all ICL-X baselines perform much better than previous fine-tuning approaches for the 0-CLT task. This reaffirms the importance of studying and improving in-context learning over very large language models for our setting.

Comparing among SSP variants, it is not surprising that GPT-4 performance supercedes GPT-3.5, which is much better than Llama2 70B. We next compare ICL baselines and SSP variants, when using the same LLM. We find that SSP's two stage workflow consistently outperforms ICL by significant margins. In fact, in-language exemplars with very noisy labels from stage 1 (E.g. for Got language with GPT-3.5-Turbo) perform quite well. These observations underscore the value of target language exemplars in ICL, even at the cost of label noise.

Finally, we compare SSP with Stage I via ICL over an LLM vs. via a fine-tuning baseline (ZGUL or mDeBerta). Fine-tuning baseline for Stage I has two benefits – it is cheaper (due to no LLM calls in Stage I), and has prediction confidence that can allow ILP to select highly confident Stage II exemplars. Due to the latter, in two of the three language groups, the use of a fine-tuning baseline performs much better, and in the third group, it is marginally behind due to weaker performance in one language (Gothic). This happens because ZGUL has a particularly poor performance on this language, leading to much noisier labels in Stage II exemplars.

Overall, our best SSP solution uses a fine-tuning baseline (ZGUL or mDeBerta) for Stage I and GPT-4 for Stage II, using its novel ILP-based exemplar selection. It outperforms closest baselines by around 3 F1 pts, on average, establishing a new state of the art for zero-shot cross lingual transfer to low resource languages. The best SSP reported results are statistically significant compared to the best baseline using McNemar's test (p-values in Tables 1 and 2 captions). We believe that our work is a significant advancement to the existing paradigm (Tanwar et al., 2023; Nambi et al., 2023), which is restricted to optimizing only one round of in-context learning. More detailed analysis on this follows in Appendix E.

5.1 Ablation Study

We now discuss the results of removing ILP components in Stage II exemplar selection. Tables 1, and 4 (last four rows) report the impact of removing confidence thresholding constraint, label coverage constraint, both of these constraints (i.e., just using similarity) from the ILP. The final row removes ILP completely and presents results of random exemplars in Stage II. All these ablations are done on SSP with ZGUL/mDeBerta for Stage I, as only those output prediction probabilities.

Impact of label coverage: We observe an average gain of 1.3 F1 points over AmericasNLI task compared to the ablation model that does not ensure label coverage as a constraint. To investigate further, we compute the average number of exemplars for each label that are covered in the selected set for both methods, along with their label-wise F1 scores (see Figure 3). We observe that the 'neutral' label is not sampled in most cases for *w/o label coverage* variant, while exactly one 'neutral' label is sampled in the SSP(mDeBerta), with label constraint. We find that this happens as the smaller fine-tuned model mDeBerta-CL has very poor recall (0.24) for 'neutral' class and hence any selection strategy has a natural tendency to not sample this label, unless enforced via a constraint. The class-wise recall scores for SSP(DeBerta^{CL})-GPT4 with and without label coverage are presented in Table 7. We observe a difference of 22 recall points for 'neutral' class (57.6 vs 35.6) between the two

Model	Neu.	Ent.	Con.	Macro-F1
DeBerta^{CL}	34.7	53	40.3	42.6
SSP-V2	**51.7**	**53.4**	51.4	**52.2**
(w/o Label)	42.6	52.3	**57.9**	50.9

Table 5: Labelwise F1 scores for fine-tuned model (DeBerta-CL) and SSP-V2 variants w. and w/o Label coverage (GPT-4-Turbo)

ILP variants. An example illustrating this behavior in terms of the exemplars selected by both methods is shown in Figure 6 (appendix).

Impact of confidence thresholding: For sequence labeling tasks, confidence thresholding plays a key role. This is validated from ablation results in Table 1, wherein removing confidence thresholding constraint from SSP leads to 5.7 points drop for POS tagging (Germanic) and 1.3 points for NER. The drop is particularly significant (around 13.5 F1 points) for Gothic (Got), which shows that not utilizing the confidence scores can lead to drastic drop. This may be because performance of ZGUL is already poor on Gothic (21 F1 points), but confidence thresholding may have likely compensated by picking higher quality exemplars. Removing thresholding would increase noise in exemplars considerably, leading to the drop.

We further study its impact by computing the quality of Stage II exemplars selected by SSP(mDeBerta), as well as all it's ablation variants. We compute the label-wise precision over all K×N (K=8, N=no. of test instances) samples for each target language, and then report their macro-average. We observe for (Figure 3) that the macro-precision of selected exemplars by the complete ILP is consistently higher than it's other ablation variants, the least value being of w/o both (similarity-based) variant. This implies that the ILP is able to effectively sample high-precision exemplars which, in turn, gets translated into it's superior downstream performance on the task.

For completeness, we also show the exemplar precision statistics for NER and POS (averaged over their label-wise precision scores) in Figure 4. The trends hold similar in the sense-that 'w/o confidence' and 'similarity-based' variants have significantly lower precision than SSP. This is expected because both these eschew confidence thresholding, leading to sampling of lower-confidence predictions. This translates to worse downstream performance (see Table 1). On the other hand, the 'w/o label coverage' variant is competitive in terms of

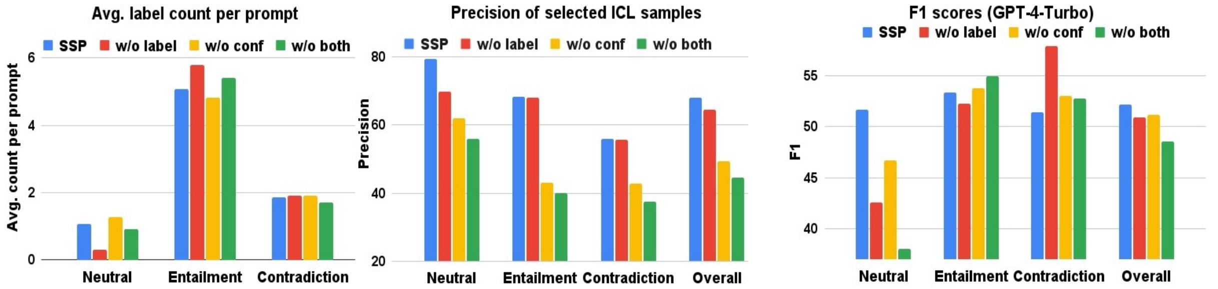

Figure 3: Label-wise statistics for AmericasNLI: Left to right - Label-wise count per prompt, Precision of ICL exemplars, and F1 results (averaged over target languages) using different selection strategies (GPT-4-Turbo)

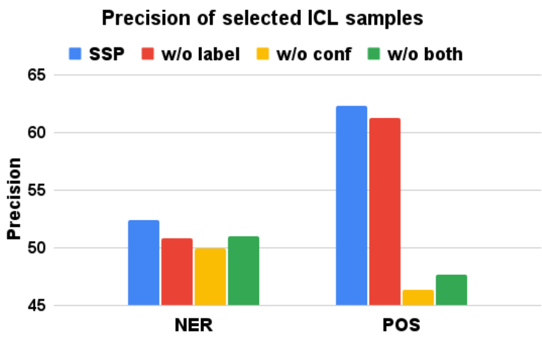

Figure 4: Precision of selected exemplars for African NER and Germanic POS

both exemplars' precision as well as downstream performance for sequence labeling tasks. This so happens, because in sequence labeling, the issue of label coverage hardly matters, since as many labels as words in the ICL set are covered in the prompt, unlike classification tasks in which only K (in our case, 8) labels can be selected.

We also note that w/o ILP (completely random selection) ablation performs much worse than SSP, showcasing the importance of carefully selecting the exemplar set.

5.2 Error Analysis

We investigate scenarios where SSP approach systematically fails compared to other methods. For NER, we find that ZGUL (fine-tuned LM) underpredicts the 'DATE' label. As a result, SSP almost never samples this label in stage 2 exemplars, hence hurting the performance for this label. For NLI task, we observe that in order to ensure label coverage, SSP samples the underpredicted label 'neutral' but while doing so, also ends up hurting the performance for 'contradiction' label (as seen in last plot of Figure 3).

6 Conclusions and Future Work

We study the zero-shot cross-lingual transfer setting for low-resource languages, when task-specific training data is available for related medium resource languages. We present Self-Supervised Prompting (SSP) – a novel two-stage framework for the use of in-context learning over very large language models. At a high-level, SSP first noisily labels the target test set using source training data (either by training a model/adapter) or by in-context learning over an LLM. SSP then uses these noisily labeled target data points as exemplars in in-context learning over the LLM. A key technical contribution is the use of integer-linear program that balances exemplar similarity, labeling confidence and label coverage to select the exemplars for a given test point. Thorough experiments on three NLP tasks, and twelve low-resource languages from three language groups show strongly improved performance over published baselines, obtaining a new state of the art in the setting. Ablations show the value each ILP component in downstream performance.

In the future, we seek to extend our technique to more non-trivial applications such as cross-lingual generation and semantic parsing. We also posit that smaller fine-tuned models, when calibrated properly, can result in more efficient selection of exemplars to an LLM, as compared to poorly calibrated counterparts, in terms of downstream performance. We leave a careful and systematic investigation into this hypothesis for future work. Moreover, we currently cover the languages having Roman scripts only, but, we seek to extend our work for non-Roman script languages as well.

7 Limitations

We show all our results and ablations on the recent state-of-the-art LLMs including GPT4. The inference for these LLMs is expensive, and makes the model deployment infeasible. Other potential limitations are extending our method to tasks such as fact checking, in which the LLMs suffer from *hallucinations* and overprediction issues. The reason why we don't use LLM logits in ILP framework is because they are not openly released by OpenAI and hence, there becomes a need to rely on smaller fine-tuned models - which can potentially lead to sub-optimal downstream performance, in case the fine-tuned models are poorly calibrated. Another serious implication of using LLMs for non-roman script languages is unreasonably high *fertility* (tokens per word split) of the LLM tokenizers, which increases the cost as well as strips the input prompt, which is not desirable.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

David Ifeoluwa Adelani, Jade Abbott, Graham Neubig, Daniel D'souza, Julia Kreutzer, Constantine Lignos, Chester Palen-Michel, Happy Buzaaba, Shruti Rijhwani, Sebastian Ruder, et al. 2021. Masakhaner: Named entity recognition for african languages. *Transactions of the Association for Computational Linguistics*, 9:1116–1131.

Sweta Agrawal, Chunting Zhou, Mike Lewis, Luke Zettlemoyer, and Marjan Ghazvininejad. 2022. In-context examples selection for machine translation. In *Annual Meeting of the Association for Computational Linguistics*.

Kabir Ahuja, Rishav Hada, Millicent Ochieng, Prachi Jain, Harshita Diddee, Samuel Maina, Tanuja Ganu, Sameer Segal, Maxamed Axmed, Kalika Bali, et al. 2023. Mega: Multilingual evaluation of generative ai. *arXiv preprint arXiv:2303.12528*.

Kabir Ahuja, Sunayana Sitaram, Sandipan Dandapat, and Monojit Choudhury. 2022. On the calibration of massively multilingual language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4310–4323, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jesujoba O Alabi, David Ifeoluwa Adelani, Marius Mosbach, and Dietrich Klakow. 2022. Adapting pre-trained language models to african languages via multilingual adaptive fine-tuning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4336–4349.

Akari Asai, Sneha Kudugunta, Xinyan Velocity Yu, Terra Blevins, Hila Gonen, Machel Reid, Yulia Tsvetkov, Sebastian Ruder, and Hannaneh Hajishirzi. 2023. Buffet: Benchmarking large language models for few-shot cross-lingual transfer. *arXiv preprint arXiv:2305.14857*.

L. Bergroth, H. Hakonen, and T. Raita. 2000. A survey of longest common subsequence algorithms. In *Proceedings Seventh International Symposium on String Processing and Information Retrieval. SPIRE 2000*, pages 39–48.

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. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*.

Abteen Ebrahimi, Manuel Mager, Arturo Oncevay, Vishrav Chaudhary, Luis Chiruzzo, Angela Fan, John Ortega, Ricardo Ramos, Annette Rios Gonzales, Ivan Meza-Ruiz, et al. 2022. Americasnli: Evaluating zero-shot natural language understanding of pretrained multilingual models in truly low-resource languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6279–6299.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Moritz Laurer, Wouter van Atteveldt, Andreu Salleras Casas, and Kasper Welbers. 2022. Less Annotating, More Classifying – Addressing the Data Scarcity Issue of Supervised Machine Learning with Deep Transfer Learning and BERT - NLI. *Preprint*. Publisher: Open Science Framework.

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 *EMNLP*.

Benjamin Muller, Antonios Anastasopoulos, Benoît Sagot, and Djamé Seddah. 2021. When being unseen from mbert is just the beginning: Handling new languages with multilingual language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 448–462.

Akshay Nambi, Vaibhav Balloli, Mercy Ranjit, Tanuja Ganu, Kabir Ahuja, Sunayana Sitaram, and Kalika Bali. 2023. Breaking language barriers with a leap: Learning strategies for polyglot llms. *arXiv preprint arXiv:2305.17740*.

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

Siqi Ouyang, Rong Ye, and Lei Li. 2022. On the impact of noises in crowd-sourced data for speech translation. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 92–97, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. Mad-x: An adapter-based framework for multi-task cross-lingual transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673.

Vipul Rathore, Rajdeep Dhingra, Parag Singla, et al. 2023. Zgul: Zero-shot generalization to unseen languages using multi-source ensembling of language adapters. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6969–6987.

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 (EMNLP)*, pages 4512–4525.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2023. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. *ArXiv*, abs/2310.11324.

Eshaan Tanwar, Subhabrata Dutta, Manish Borthakur, and Tanmoy Chakraborty. 2023. Multilingual LLMs are better cross-lingual in-context learners with alignment. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6292–6307, Toronto, Canada. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ahmet Üstün, Arianna Bisazza, Gosse Bouma, and Gertjan van Noord. 2020. Udapter: Language adaptation for truly universal dependency parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2302–2315.

Anton Voronov, Lena Wolf, and Max Ryabinin. 2024. Mind your format: Towards consistent evaluation of in-context learning improvements. *arXiv preprint arXiv:2401.06766*.

Xingchen Wan, Ruoxi Sun, Hootan Nakhost, Hanjun Dai, Julian Eisenschlos, Sercan Arik, and Tomas Pfister. 2023. Universal self-adaptive prompting. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7437–7462, Singapore. Association for Computational Linguistics.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, and Tengyu Ma. 2023. Larger language models do in-context learning differently.

Genta Indra Winata, Andrea Madotto, Zhaojiang Lin, Rosanne Liu, Jason Yosinski, and Pascale Fung. 2021. Language models are few-shot multilingual learners. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 1–15.

Ningyu Zhang, Luoqiu Li, Xiang Chen, Shumin Deng, Zhen Bi, Chuanqi Tan, Fei Huang, and Huajun Chen. 2021. Differentiable prompt makes pre-trained language models better few-shot learners. *arXiv preprint arXiv:2108.13161*.

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

A Implementation and Hyperparameter Details

We use Azure OpenAI service [3] for all experiments involving GPT-3x and GPT-4x models. For LLama-2-70b, we use the together API [4]. We set temperature as 0.0 consistently for all our experiments, making our results directly reproducible. The max_tokens (max. no. of generated tokens) parameter is set to 1024 for POS and NER tasks, while 15 for the NLI. For all experiments, the no. of exemplars (M) is fixed to 8 for uniform comparison. For ILP solver, we use Python's gurobipy [5] package.

A.1 Estimating confidence $\hat{y}_k^i$

For NLI task, the model always predicts a single label: 'neutral', 'contradiction' or 'entailment'. We simply apply softmax on the class logits for the predicted label to compute the confidence $\hat{y}_j^i$ (for i^{th} test instance).

In sequence labeling tasks, suppose for an input sentence having words: $\{w_1, w_2, ..., w_T\}$, the model predicts labels $\{o_1, o_2, ..., o_T\}$ with probabilities $\{\hat{p}_1, \hat{p}_2, ..., \hat{p}_T\}$. Let $LabelSet$ be $\{l_1, l_2, ..., l_N\}$. We compute confidence $\hat{y}_l$ for each label for a given test example as follows:

for $k \leftarrow 1$ to N **do**
 $\hat{y}_k \leftarrow 0$ ▷ init each label's confidence
 $c_k \leftarrow 0$ ▷ init each label's count
end for
for $i \leftarrow 1$ to T **do**
 for $j \leftarrow 1$ to N **do**
  if $l_j == o_i$ **then**
   $\hat{y}_j \leftarrow \hat{y}_j + \hat{p}_i$ ▷ Update $\hat{y}_j$
   $c_j \leftarrow c_j + 1$ ▷ increase counter
  end if
 end for
end for
for $k \leftarrow 1$ to N **do**
 $\hat{y}_k = \hat{y}_k / c_k$ ▷ average over all occurrences
end for

This outputs the confidence scores $\hat{y}_l$ for a given example, with those not predicted in a sequence having 0 value.

[3] https://azure.microsoft.com/en-in/products/ai-services/openai-service

[4] https://www.together.ai/

[5] https://pypi.org/project/gurobipy/

B Prompt details

Prompts for the Named Entity Recognition (NER) and Part of Speech Tagging (POS) tasks are presented in the tab separated format shown in B.0.2 and B.0.3 respectively.

Prompts for Natural Language Inference (NLI) initially used the framework in Ahuja et al. (2023) . To improve our performance, we changed the prompt to use Sclar et al. (2023)'s framework, where the authors performed an exhaustive search over tokens used for a prompt in order to find the prompt with optimal performance. This increased Macro F1 score by atleast 10% across all the tested languages. We use the same prompt across all models used in our experiments.

B.0.1 Natural Language Inference (NLI)

Task Description: You are an NLP assistant whose purpose is to solve Natural Language Inference (NLI) problems. NLI is the task of determining the inference relation between two (short, ordered) texts: entailment, contradiction, or neutral. Answer as concisely as possible in the same format as the examples below:
Input format:
Premise: {premise} , Hypothesis: {hypothesis} ,
Output format:
Answer: {output}
Verbalizer:
match the one-word response from the model (neutral, contradiction or entailment)

B.0.2 Named Entity Recognition (NER)

Task Description: Tag the following sentence according to the BIO scheme for the NER task, using the tags PER (person), LOC (location), ORG (organization) and DATE (date). Follow the format specified in the examples below:
Input format:
Sentence: w_1 w_2 ... w_T
Output format:
Tags:
w_1<TAB>o_1
w_2<TAB>o_2
...
w_T<TAB>o_T
Verbalizer:
Extract the sequence of labels $o_1, o_2, ...o_3$ from generated response.

B.0.3 Part of Speech (PoS) tagging

Task Description: Tag the following sentence according to the Part of Speech (POS) of each word. The valid tags are ADJ, ADP, ADV, AUX, CCONJ, DET, INTJ, NOUN, NUM, PART, PRON, PROPN, PUNCT, SCONJ, SYM, VERB, X. Follow the format specified in the examples below:
Input format:
Sentence: w_1 w_2 ... w_T
Output format:
Tags:
w_1<TAB>o_1
w_2<TAB>o_2
...
w_T<TAB>o_T
Verbalizer:
Extract the sequence of labels $o_1, o_2, ...o_3$ from generated response.

B.1 Verbalizer details for Tagging tasks

The verbalizer for tagging tasks requires the LLM to output the words as well as the associated labels. The LLM's output may not be perfect, as it may fail to generate all words or associate a label with each word. As a result, we find the *Longest Common Subsequence* between the words generated by the LLM and the words of the example. This is done using Dynamic Programming, as described in (Bergroth et al., 2000).

Once we have found the longest common subsequence, we assign the corresponding tags generated by the LLM to these words. If the tags are invalid, we assign a default tag (O for NER, and X for POS). Finally, for the words which don't have any tags associated with them, we assign the same default tag as before.

It is to be noted that in most cases, the sentence generated by the LLM perfectly matches the original sentence. For GPT-4, less than 1% of the words fell into the category of having an invalid tag generated, or not having the word generated.

B.2 Prompts for GSW Examples

The base SSP-SIM prompts for the GSW examples highlighted in Figure 5 are given below. Labels which are misclassified in the in-context exemplars are coloured in red, and the AUX labels which are to be flipped in the ablations are coloured in blue. It is interesting to note that examples 1 and 2 are similar, as example 1 is retrieved as an in-context exemplar for example 2.

B.2.1 Example 1

Tag the following sentence according to the Part of Speech (POS) of each word. The valid tags are ADJ, ADP, ADV, AUX, CCONJ, DET, INTJ, NOUN, NUM, PART, PRON, PROPN, PUNCT, SCONJ, SYM, VERB, X. Follow the format specified in the examples below:
Sentence: I main , das Ganze letscht Wuchä isch mier scho ächli iigfaarä .
Tags:
“‘
I PRON
main VERB
, PUNCT
das DET
Ganze NOUN
letscht ADJ
Wuchä NOUN
isch AUX
mier PRON
scho ADV
ächli ADV
iigfaarä VERB
. PUNCT
“‘
Sentence: Du gsehsch uus , wi wenn de nöime no hättisch z trinken übercho .
Tags:
“‘
Du PRON
gsehsch VERB
uus PRON
, PUNCT
wi SCONJ
wenn SCONJ
de DET
nöime ADJ
no ADV
hättisch AUX
z PART
trinken VERB
übercho VERB
. PUNCT
“‘
Sentence: Dir weit mer doch nid verzöue , di Wäutsche heige vo eim Tag uf en anger ufghört Chuttlen ässe .
Tags:
“‘
Dir PRON
weit VERB

	Ds	Gueten	isch	immerhin	gsi	,	dass	i	ungerdesse	söfu	müed	bi	gsi	,	dass	i	ändlech	ha	chönne	go	schlofe	.
CLT-SIM	DET	NOUN	AUX	ADV	VERB	PUNCT	SCONJ	PRON	ADV	VERB	ADJ	ADP	VERB	PUNCT	SCONJ	PRON	ADV	AUX	AUX	VERB	VERB	PUNCT
SSP-CLT-SIM	DET	NOUN	AUX	ADV	AUX	PUNCT	SCONJ	PRON	ADV	ADV	ADJ	ADP	AUX	PUNCT	SCONJ	PRON	ADV	AUX	AUX	PART	VERB	PUNCT
SSP-CLT-SIM (Half AUX->VERB)	DET	NOUN	AUX	ADV	AUX	PUNCT	SCONJ	PRON	ADV	ADV	ADJ	ADP	AUX	PUNCT	SCONJ	PRON	ADV	AUX	AUX	PART	VERB	PUNCT
SSP-CLT-SIM (All AUX->VERB)	DET	NOUN	VERB	ADV	VERB	PUNCT	SCONJ	PRON	ADV	ADV	ADJ	ADP	VERB	PUNCT	SCONJ	PRON	ADV	AUX	AUX	VERB	VERB	PUNCT
Gold	DET	NOUN	AUX	ADV	AUX	PUNCT	SCONJ	PRON	ADV	ADV	ADJ	AUX	AUX	PUNCT	SCONJ	PRON	ADV	AUX	AUX	PART	VERB	PUNCT

	I	cha	der	ihri	Telefonnummere	gä	,	de	nimmsch	mou	unverbindlech	Kontakt	uuf	.
CLT-SIM	PRON	VERB	DET	ADJ	NOUN	VERB	PUNCT	PRON	VERB	ADV	ADJ	NOUN	VERB	PUNCT
SSP-CLT-SIM	PRON	AUX	PRON	PRON	NOUN	VERB	PUNCT	PRON	VERB	ADV	ADJ	NOUN	ADP	PUNCT
SSP-CLT-SIM (Half AUX->VERB)	PRON	AUX	PRON	PRON	NOUN	VERB	PUNCT	PRON	VERB	ADV	ADJ	NOUN	ADP	PUNCT
SSP-CLT-SIM (All AUX->VERB)	PRON	VERB	PRON	PRON	NOUN	VERB	PUNCT	DET	VERB	ADV	ADJ	NOUN	ADP	PUNCT
Gold	PRON	AUX	PRON	DET	NOUN	VERB	PUNCT	ADV	VERB	ADV	ADJ	NOUN	PART	PUNCT

Figure 5: Label flips for CLT-SIM and SSP-SIM, for POS tagging in Swiss-German (gsw). Incorrect labels are marked in red. SSP-SIM ablations include flipping half/all of the AUX labels in the prompt to VERB labels. Gold labels are given for reference.

mer PRON
doch ADV
nid ADV
verzöue VERB
, PUNCT
di DET
Wäutsche NOUN
heige VERB
vo ADP
eim DET
Tag NOUN
uf ADP
en DET
anger ADJ
ufghört VERB
Chuttlen NOUN
ässe VERB
. PUNCT
‘‘‘

Sentence: es isch nämli echt usgstorbe gsi .
Tags:
‘‘‘

es PRON
isch AUX
nämli ADV
echt ADJ
usgstorbe VERB
gsi AUX
. PUNCT
‘‘‘

Sentence: Aso bini rächt uufgschmissä gsi und dem entschprächend fascht verzwiiflät .
Tags:
‘‘‘

Aso ADV
bini AUX
rächt ADV
uufgschmissä VERB
gsi AUX
und CCONJ
dem PRON
entschprächend ADJ
fascht ADV
verzwiiflät VERB
. PUNCT
‘‘‘

Sentence: Der Ääschme wett nöd schaffe biin em .
Tags:
‘‘‘

Der DET
Ääschme NOUN
wett AUX
nöd ADV
schaffe VERB
biin ADP
em PRON
. PUNCT
‘‘‘

Sentence: Zerscht hends am Dani gsait , är söli dòch Hoochdütsch redä , das gängi denn grad gaar nöd , wenn är so redi , wiäner redi .
Tags:
‘‘‘

Zerscht ADV
hends PRON
am ADP
Dani PROPN
gsait VERB
, PUNCT
är PRON
söli AUX
dòch ADV
Hoochdütsch ADJ
redä VERB

, PUNCT
das PRON
gängi VERB
denn ADV
grad ADV
gaar ADV
nöd ADV
, PUNCT
wenn SCONJ
är PRON
so ADV
redi VERB
, PUNCT
wiäner PRON
redi VERB
. PUNCT
```
Sentence: Isch das e Sach gsi , bis mer se gfunge hei gha .
Tags:
```
Isch AUX
das PRON
e DET
Sach NOUN
gsi AUX
, PUNCT
bis SCONJ
mer PRON
se PRON
gfunge VERB
hei AUX
gha VERB
. PUNCT
```
Sentence: Ds Gueten isch immerhin gsi , dass i ungerdesse söfu müed bi gsi , dass i ändlech ha chönne go schlofe .
Tags:
```

B.2.2 Example 2

Tag the following sentence according to the Part of Speech (POS) of each word. The valid tags are ADJ, ADP, ADV, AUX, CCONJ, DET, INTJ, NOUN, NUM, PART, PRON, PROPN, PUNCT, SCONJ, SYM, VERB, X. Follow the format specified in the examples below:
Sentence: I ha ar Marie-Claire gseit , es sig mer chli schlächt und i mög jetz nümm liire .
Tags:
```
I PRON
ha AUX
ar PART
Marie-Claire PROPN
gseit VERB
, PUNCT
es PRON
sig AUX
mer PRON
chli ADV
schlächt ADJ
und CCONJ
i PRON
mög VERB
jetz ADV
nümm ADV
liire VERB
. PUNCT
```
Sentence: De Spanier hed de Kontakt vermettlet , d Rumäne sölled d Holländer ombrocht ha .
Tags:
```
De DET
Spanier NOUN
hed AUX
de DET
Kontakt NOUN
vermettlet VERB
, PUNCT
d DET
Rumäne NOUN
sölled AUX
d DET
Holländer PROPN
ombrocht VERB
ha AUX
. PUNCT
```
Sentence: Ds Gueten isch immerhin gsi , dass i ungerdesse söfu müed bi gsi , dass i ändlech ha chönne go schlofe .
Tags:
```
Ds DET
Gueten NOUN
isch AUX
immerhin ADV
gsi VERB
, PUNCT
dass SCONJ
```

i PRON
ungerdesse ADV
söfu VERB
müed ADJ
bi ADP
gsi VERB
, PUNCT
dass SCONJ
i PRON
ändlech ADV
ha AUX
chönne AUX
go VERB
schlofe VERB
. PUNCT
‘‘‘

Sentence: Isch das e Sach gsi , bis mer se gfunge hei gha .
Tags:
‘‘‘
Isch AUX
das PRON
e DET
Sach NOUN
gsi AUX
, PUNCT
bis SCONJ
mer PRON
se PRON
gfunge VERB
hei AUX
gha VERB
. PUNCT
‘‘‘

Sentence: De Dialäkt muess zu de Gschecht und zum Inhaut vonere Werbig passe .
Tags:
‘‘‘
De DET
Dialäkt NOUN
muess AUX
zu ADP
de DET
Gschecht NOUN
und CCONJ
zum ADP
Inhaut NOUN
vonere ADP
Werbig NOUN
passe VERB
. PUNCT
‘‘‘

Sentence: Mit der Zit hani mi mit mir säuber uf ei Schriibwiis pro Wort aafo einige .
Tags:
‘‘‘
Mit ADP
der DET
Zit NOUN
hani VERB
mi PRON
mit ADP
mir PRON
säuber ADJ
uf ADP
ei DET
Schriibwiis NOUN
pro ADP
Wort NOUN
aafo VERB
einige DET
. PUNCT
‘‘‘

Sentence: Mit all denä Wörter hani natürli nüt chönä aafangä .
Tags:
‘‘‘
Mit ADP
all DET
denä DET
Wörter NOUN
hani PRON
natürli ADV
nüt ADV
chönä VERB
aafangä VERB
. PUNCT
‘‘‘

Sentence: Aso bini rächt uufgschmissä gsi und dem entschprächend fascht verzwiiflät .
Tags:
‘‘‘
Aso ADV
bini AUX
rächt ADV
uufgschmissä VERB
gsi AUX
und CCONJ
dem PRON
entschprächend ADJ
fascht ADV
verzwiiflät VERB
. PUNCT
‘‘‘

Model	Neu.	Ent.	Con.	Overall
DeBerta^{CL}	24.3	**72.7**	38.7	45.2
SSP-V2	**57.8**	46.5	51.5	52
(w/o Label)	35.3	43.8	68.5	49.2

Table 7: Labelwise Recall for fine-tuned model (DeBerta-based) and ILP variants w. and w/o Label coverage (GPT-4-Turbo)

Sentence: I cha der ihri Telefonnummere gä , de nimmsch mou unverbindlech Kontakt uuf .
Tags:
‘‘‘

C Source and Target Languages for each task

Code	Language
En	English
Am	Amharic
Sw	Swahili
Wo	Wolof
Hau	Hausa
Ibo	Igbo
Kin	Kinyarwanda
Lug	Luganda
Luo	Luo
Is	Icelandic
De	German
Fo	Faroese
Got	Gothic
Gsw	Swiss German
Nds	Low-Saxon
Es	Spanish
Aym	Aymara
Gn	Guarani
Nah	Nahuatl

Table 6: Languages and their codes

D NLI Label coverage Analysis

We present an example of correct prediction made by SSP as compared to the version that doesn't ensure label coverage in Figure 6 (English translation in Fig. 7).

E Qualitative Analysis: SSP-SIM

We present the analysis for the gains obtained via SSP-SIM for Germanic POS in Figure 8. The confusion matrix difference between SSP-SIM and CLT-SIM suggests that the model misclassifies auxiliary verbs as verbs in CLT-SIM, and this is corrected in SSP-SIM. These errors are a consequence of the labels on the in-context exemplars the model receives, and not the tokens of the language itself.

We highlight this via the two Swiss-German POS examples in Figure 5. The misclassified verbs are corrected by SSP-SIM, and these labels are again misclassified when more than half of the labels in the in-context exemplars are corrupted.

F Data Contamination Analysis

Following Ahuja et al. 2023, we conduct contamination tests on test datasets for our target languages. We perform the following tests:

- Dataset Card filling: Generate dataset card (supported languages, dataset description, #instances in each split, etc.)
- Completion: Given a few words, complete the sentence and their labels, and
- Generation using first few instances: Given first K instances (K=5) in the dataset, generate next few instances following them.

We observe negligible contamination as depicted in table 8. The 40% accuracy for Quechua was a result of all the labels passed for the exemplars being entailment labels. As a result, the model repeated the same label for all the other examples, giving a 40% accuracy. *Following these results, to prevent any chance of contamination, we remove Quechua from our evaluation dataset.*

Premise: Ah, huk chaypi allinqa apakurqa allin qawasqayqa paniypa ñawpaq yuyariyninmi, chaypas hina hipa pampapim karqa.
Hypothesis: Yuyaruniqa hipa pampapi huk ima apakusqantam.
Answer: entailment

Premise: Yaykuykuptiykuqa punkukunaqa wichqasqam kachkarqa.
Hypothesis: Punku wichqasqa kachkaptinpas yaykurqanikum.
Answer: entailment

Premise: Yanapawaqniy atiq sispasmi hatun llaqtapa waklawninpiraq tiyan.
Hypothesis: Yanapawaqniy warmi warman 5 millas nisqan karupiraq tiyan.
Answer: **neutral**

Premise: Manam mayman risqanta yacharqanikuchu.
Hypothesis: Mayman risqantam yacharqaniku.
Answer: entailment

Premise: Chayna kaptinqa hamutachkanim huktapiwan Ramonawan rimariyta.
Hypothesis: Ramonawanmi huktapiwan rimarqani.
Answer: entailment

Premise: Ripukusqañam hinaspam amaña llakikunaypaq niwarqa.
Hypothesis: Ama llakikunaytam niwarqa.
Answer: entailment

Premise: Ichapasyá huk kaq mana yachasqaymanta hamun ichaqa
Hypothesis: Apurawtam hamun, ichaqa maymanta hamusqanta yachanim.
Answer: entailment

Premise: Locust Hill oh awriki, ari, kusa
Hypothesis: Locust Hill nisqaqa allinmi.
Answer: contradiction

Premise: Oh, payllam isqun iskay iskayraq regulador nisqapi inyecciónta qinaq karqa.
Hypothesis: Martes punchawtam inyector nisqata hinarqani.
Answer: **neutral**

Premise: Ah, huk chaypi allinqa apakurqa allin qawasqayqa paniypa ñawpaq yuyariyninmi, chaypas hina hipa pampapim karqa.
Hypothesis: Yuyaruniqa hipa pampapi huk ima apakusqantam.
Answer: entailment

Premise: Yaykuykuptiykuqa punkukunaqa wichqasqam kachkarqa.
Hypothesis: Punku wichqasqa kachkaptinpas yaykurqanikum.
Answer: entailment

Premise: Manam mayman risqanta yacharqanikuchu.
Hypothesis: Mayman risqantam yacharqaniku.
Answer: entailment

Premise: Chayna kaptinqa hamutachkanim huktapiwan Ramonawan rimariyta.
Hypothesis: Ramonawanmi huktapiwan rimarqani.
Answer: entailment

Premise: Manam pachay karqachu ima kaqpas ruranaypaq.
Hypothesis: Mana pacha llapan qinanaypaq haypawarqachu
Answer: entailment

Premise: Ripukusqañam hinaspam amaña llakikunaypaq niwarqa.
Hypothesis: Ama llakikunaytam niwarqa.
Answer: entailment

Premise: Ichapasyá huk kaq mana yachasqaymanta hamun ichaqa
Hypothesis: Apurawtam hamun, ichaqa maymanta hamusqanta yachanim.
Answer: entailment

Premise: Locust Hill oh awriki, ari, kusa
Hypothesis: Locust Hill nisqaqa allinmi.
Answer: contradiction

Premise: Oh, payllam isqun iskay iskayraq regulador nisqapi inyeccionta qinaq karqa. Hypothesis: Martes punchawtam inyector nisqata hinarqani.
Answer: **contradiction**

Figure 6: Correct case of 'Neutral' detected by ILP (left), while 'w/o label' variant misses it (right). We note that exact one 'neutral' class has been sampled by ILP, while no 'neutral' is sampled in 'w/o label' version.

Premise: Ah, one there good thing took away is my best view is my sister's old memory, which was also on the same hip floor.
Hypothesis: I remember something carrying on the floor.
Answer: entailment

Premise: The doors were locked when we entered.
Hypothesis: We got in even though the door was locked.
Answer: entailment

Premise: The sister who can help me lives just on the other side of the big city.
Hypothesis: My assistant lives 5 miles away.
Answer: **neutral**

Premise: We didn't know where he was going.
Hypothesis: We knew where he was going.
Answer: entailment

Premise: In that case I'm coming up with another conversation with Ramona.
Hypothesis: I talked to Ramona again.
Answer: entailment

Premise: He had left and told me not to worry.
Hypothesis: He told me not to worry.
Answer: entailment

Premise: Maybe it comes from something I don't know though
Hypothesis: It comes quickly, but I know where it comes from.
Answer: entailment

Premise: Locust Hill oh yeah, yeah, great
Hypothesis: Locust Hill is good.
Answer: contradiction

Premise: Oh, he was the only one who still injected nine seconds into the regulator.
Hypothesis: I applied the injector on Tuesday.
Answer: **neutral**

Premise: Ah, one there good thing took away is my best view is my sister's old memory, which was also on the same hip floor.
Hypothesis: I remember something carrying on the floor.
Answer: entailment

Premise: The doors were locked when we entered.
Hypothesis: We got in even though the door was locked.
Answer: entailment

Premise: We didn't know where he was going.
Hypothesis: We knew where he was going.
Answer: entailment

Premise: In that case I'm coming up with another conversation with Ramona.
Hypothesis: I talked to Ramona again.
Answer: entailment

Premise: I didn't have time to do anything.
Hypothesis: I didn't have enough time to cover everything
Answer: entailment

Premise: He had left and told me not to worry.
Hypothesis: He told me not to worry.
Answer: entailment

Premise: Maybe it comes from something I don't know though
Hypothesis: It comes quickly, but I know where it comes from.
Answer: entailment

Premise: Locust Hill oh yeah, yeah, great
Hypothesis: Locust Hill is good.
Answer: contradiction

Premise: Oh, he was the only one who still injected nine seconds into the regulator. Hypothesis: I applied the injector on Tuesday.
Answer: **contradiction**

Figure 7: English translations of Exemplars shown in Fig. 6

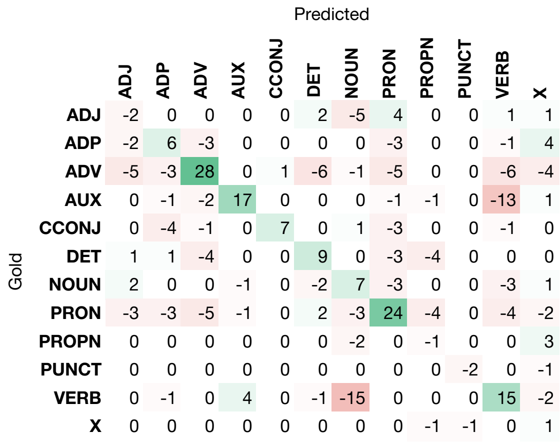

Figure 8: Difference in confusion matrices between SSP-SIM and CLT-SIM for the POS task, summed across all languages (tags with less than 100 instances have been omitted). The increase in correct tags is visible along the diagonal, and misclassifications between VERB and AUX tags / NOUN and VERB tags have also improved.

Task	Card Filling	Completion	Few-Shot Generation
NER	Didn't predict correct languages; no split sizes generated	No match found	NA
POS	predicted 33 languages, but doesn't contain any of our target languages	No match found	NA
NLI	predicts 3 languages, of which only one matches with our target language (Quechua); wrong test split size	Refuses to generate for 3 out of 4 target languages, except for Quechua - for which it predicts 100% of the tokens wrong and only 40% labels correctly (out of 10 instances)	Repeats the premise of last instance, copies the premise string to hypothesis as well (No match detected)

Table 8: Results of Contamination Study