An End-to-End Submodular Framework for Data-Efficient In-Context Learning

Lilly Kumari[‡], **Shengjie Wang**[§], **Arnav Das**[‡], **Tianyi Zhou**[¶], **Jeff Bilmes**[‡]
[‡]University of Washington, [§]NYU Shanghai, [¶]University of Maryland
{lkumari,arnavmd2,bilmes}@uw.edu, shengjie.wang@nyu.edu, zhou@umiacs.umd.edu

Abstract

Recent advancements in natural language tasks leverage the emergent In-Context Learning (ICL) ability of pretrained Large Language Models (LLMs). ICL enables LLMs to perform new tasks by utilizing a limited number of input-output examples as prompts. While ICL circumvents the costly step of finetuning LLMs, its effectiveness is heavily dependent on the quality and ordering of provided examples (called exemplars). In this work, we propose a two-stage data-efficient framework *Div-S3* for exemplar selection for ICL. The first stage focuses on data annotation and employs a pool-based active learning approach to select a set of *Div*erse and informative exemplars from the target tasks' unlabeled pool. Given a test input/query, the second stage uses Submodular Span Summarization (*S3*) to select the most relevant and non-redundant exemplars from the annotated pool of a limited budget. On 7 different NLP datasets and 5 LLMs of varying complexities, we show *Div-S3* outperforms (1) existing active learning-based methods for data annotation for ICL and (2) similarity-based methods for test query-specific exemplars retrieval.

1 Introduction

Pretrained large language models (LLMs) (Kenton and Toutanova, 2019; Brown et al., 2020; Chowdhery et al., 2022) have become foundational for a wide range of Natural Language Processing (NLP) tasks, demonstrating impressive success across various domains (Bommasani et al., 2021; Bubeck et al., 2023) through in-context learning (ICL) (Dong et al., 2022). ICL enables these pretrained LLMs to perform new tasks by using task-specific prompts containing a limited number of input-output demonstrations (also referred to as shots, exemplars, or prompts) in the natural language format. This approach facilitates deployment across different downstream tasks and reduces the need for labeled downstream training data since ICL does not require any task-specific training.

The typical ICL procedure consists of two key components: (1) Exemplar annotation and retrieval (Wu et al., 2022; Köksal et al., 2022; Liu et al., 2022): This step involves annotating and retrieving exemplars that serve as context demonstrations. (2) Prompt template crafting (Sorensen et al., 2022; Deng et al., 2022): this step involves designing a prompt template to wrap these demonstrations in a comprehensible and coherent natural language instruction.

Recent studies (Liu et al., 2022; Su et al., 2022; Margatina et al., 2023) show that providing exemplars most relevant to the current input instance is beneficial. Moreover, Zhao et al. (2021), Lu et al. (2022), and Liu et al. (2023) observe that LLMs attend more to the exemplars that are closer in the sequence to the input instance. Therefore, to achieve the best performance of ICL, the selection of exemplars and their ordering in the LLM prompt are crucial.

In practice, an extensive collection of unlabeled exemplars is easily available (e.g., posts and discussions on forums like Stack Exchange or user-generated content on social media platforms), but manually annotating all exemplars would be exceptionally costly. To annotate and select the exemplars optimally for a given target task, we follow the two-stage approach shown in Figure 1: (1) **Exemplar Annotation**: select a subset of exemplars for annotation under a fixed budget (performed only once) and (2) **Exemplar Retrieval**: identify limited-sized exemplars in an ordering that are most influential for a given input instance from the annotated subset of exemplars. Intuitively, for the first stage, we aim to find the subset with maximal diversity and least redundancy so that, given any input, we can find corresponding labeled exemplars. For the second stage, in addition to the diversity requirement similar to the first stage,

Findings of the Association for Computational Linguistics: NAACL 2024, pages 3293–3308

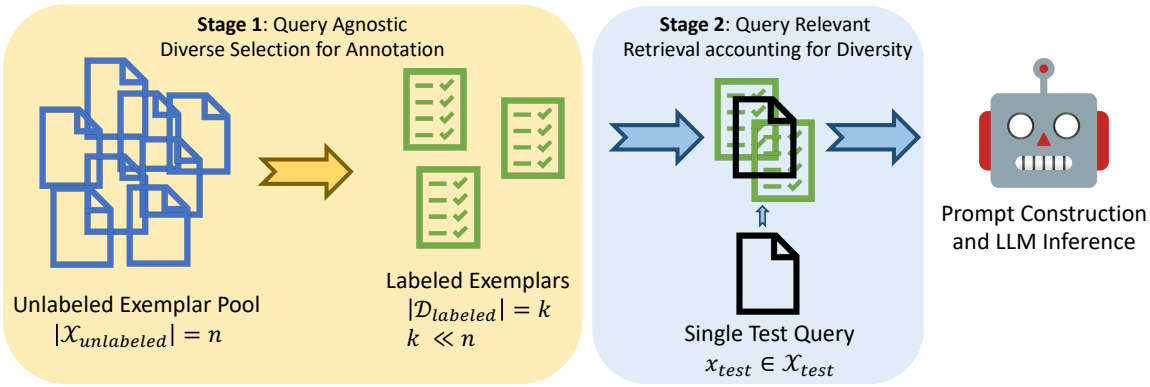


Figure 1: Workflow of our proposed framework for data-efficient In-Context Learning using LLMs. For the first stage, we use cardinality-constrained submodular optimization to identify diverse exemplars for annotation. For the second stage, given a test query, we use Submodular Span Summarization to find a diverse set of labeled exemplars that are most relevant to the query.

we emphasize the relevance of the exemplars to the given input query and order exemplars so that their relevance to the input query decreases as the exemplars are farther away from the input instance.

Test Query

occasionally melodramatic , it 's also extremely effective. (positive)

Relevance based Exemplar Retrieval (Similar)

- darkly funny and frequently insightful. (very positive)
- at once disarmingly straightforward and strikingly devious. (very positive)
- the movie is well crafted , and well executed. (very positive)
- it has plenty of laughs. (positive)
- his work with actors is particularly impressive . (very positive)
- their work is fantastic. (very positive)
- his method almost never fails him , and it works superbly here. (very positive)
- an effortlessly accomplished and richly resonant work. (very positive)

Submodular Span Summarization (S3)

- the whole affair is as predictable as can be. (neutral)
- not too fancy , not too filling , not too fluffy , but definitely tasty and sweet. (positive)
- a return to pure disney magic and is enjoyable family fare. (very positive)
- after all , it 'll probably be in video stores by christmas , and it might just be better suited to a night in the living room than a night at the movies. (negative)
- this is a more fascinating look at the future than `` bladerunner '' and one of the most high-concept sci fi adventures attempted for the screen. (positive)
- far more enjoyable than its predecessor. (very positive)
- this method almost never fails him , and it works superbly here. (very positive)
- darkly funny and frequently insightful. (very postive)

Figure 2: A sample test query from the SST-5 dataset with its corresponding set of exemplars selected using *Similar* (focusing only on relevance) and *S3* (focusing on both relevance and representativeness) from a limited exemplars pool. Exemplars are colored based on their class names. We use echo lines to denote the redundant exemplars chosen by *Similar* and used as a part of the input context during ICL.

We propose a framework *Div-S3* based on submodular optimization that unifies the above-mentioned two stages. For Exemplar Annotation, we model the problem as a submodular optimization problem under a cardinality constraint to find as **Div**erse a subset as possible within a budget. For Exemplar Retrieval, we formalize the problem as a Submodular Span Summarization (**S3**) problem (Kumari and Bilmes, 2021) with a knapsack constraint, which finds a diverse subset most relevant to the input query under a token length limit. Also, we naturally order the resulting exemplars based on the gains represented by the submodular function. The name of our proposed framework *Div-S3* captures the optimization objectives used for both exemplar annotation (*Div*) and exemplar retrieval (*S3*) stages. In Fig. 2, we show a sample test query where using *Div-S3* for exemplar selection leads to a more diverse and query-relevant exemplar set (more examples provided in Appendix D).

Our framework is general, as any submodular function can be plugged into our method. For models beyond LMs, e.g., for text-image multi-modality models, we may use pre-existing submodular functions that are powerful for expressing diversity in the image domain. In addition, we account for relevance, diversity, and ordering for the exemplar retrieval stage, where one or two aspects typically get overlooked by previous methods. Empirically, we evaluate *Div-S3* on 7 NLP tasks with 5 LLMs and show significantly improved performance compared to baselines. Our contributions are:

1. We propose an end-to-end framework *Div-S3* utilizing submodular optimization for performing data-efficient ICL using LLMs. Depending on budget requirements, *Div-S3* provides the flexibility to set the budget either in terms of the number of exemplars to be used in the prompt or the LLM's context window size.

2. We empirically validate the effectiveness of our framework on 7 different NLP tasks and show the transferability of results across LLMs of varying complexities.

3. We thoroughly analyze each component of *Div-S3* by (a) studying *S3* in a setting with no

annotation budget constraint and (b) analyzing the sensitivity of the exemplars selected by *Div-S3* to their position in the LLM's prompt.

2 Related Work

In this section, we outline recent studies investigating ICL, specifically focusing on works in the context of exemplar annotation selection and retrieval. With the introduction of ICL in the GPT-3 paper (Brown et al., 2020), numerous studies have emerged trying to understand the mechanics of ICL (Xie et al., 2021; Min et al., 2022; Chan et al., 2022; Von Oswald et al., 2023; Wei et al., 2023; Dai et al., 2023; Han et al., 2023; Li et al., 2023) and its inherent strengths and limitations (Webson and Pavlick, 2022; Lu et al., 2023; Jang et al., 2023; Kung and Peng, 2023; Yin et al., 2023).

A recent work by Min et al. (2022) shows that ICL fails to learn label relationships from the in-context exemplars, as its performance diminishes only slightly when substituting the demonstration labels with random labels. However, Kossen et al. (2023) discovers that LLMs indeed utilize the in-context label information, and label relationships learned during the pre-training phase have a more lasting effect than in-context demonstrations. Also, prior works (Zhao et al., 2021; Liu et al., 2022; Lu et al., 2022) have demonstrated the sensitivity of ICL to the choice and order of in-context exemplars in the final LLM prompt. This has prompted investigations into determining which samples should be included in the exemplar pool and how to select exemplars at test time effectively.

2.1 Active Learning for Exemplar Annotation

Recent works such as Su et al. (2022), Zhang et al. (2022b), and Köksal et al. (2022) have explored ICL under a fixed annotation budget. VoteK (Su et al., 2022) uses a combination of graph-based and uncertainty sampling-based approaches to select diverse samples for annotation and relies on the model's confidence for the uncertainty sampling. Furthermore, Zhang et al. (2022b) formulate exemplar selection as an iterative decision problem and propose a reinforcement learning algorithm to train policies for active exemplar selection.

DataModels proposed in Chang and Jia (2023) trains a linear regression model to predict the LLM's outcome given an exemplar and its position in the final prompt. The CondAcc method proposed in the same work scores each exemplar by its dev-set ICL performance when combined with other randomly sampled in-context exemplars. However, both methods require multiple rounds of inference using the target LLM and can be pretty costly in practice. Another recent work (Margatina et al., 2023) highlights that uncertainty sampling for selecting annotated exemplars results in inferior performance. In contrast, similarity-based sampling performs better, albeit with the drawback of increasing the annotation budget, as each test query is considered independently.

To find supporting examples for ICL, Li and Qiu (2023) employs a two-stage framework: (1) a progressive filtering stage, which extracts informative examples via a new metric based on LLMs' feedback, and (2) a diversity-guided beam search method to select the final supporting examples.

2.2 Exemplar Retrieval

This section primarily covers prior works studying exemplar retrieval at test time. In ICL methods that do not involve any fine-tuning, the most explored exemplar retrieval method relies on a simple cosine similarity-based ranking (Rubin et al., 2021; Liu et al., 2022; Su et al., 2022; Margatina et al., 2023; Wu et al., 2023). However, this approach treats each exemplar independently and does not capture their interactions.

Amongst the learning-based methods, Rubin et al. (2021) trains a lightweight retriever model using a contrastive learning objective to score each exemplar independently. CEIL proposed in Ye et al. (2023) similarly trains a retriever by utilizing a DPP to score a subset of exemplars. They show that selecting a set of diverse and non-redundant exemplars is vital for the overall performance of ICL. However, to learn the DPP retriever, CEIL needs to create training data by scoring multiple sets of exemplars formatted in a prompt template using the inference LLM, thus adding to the computational costs besides the costs associated with fine-tuning.

3 Notations & Background

We now outline the notations related to in-context learning using LLMs and provide some background on submodular functions and optimization.

3.1 Notations related to ICL

ICL as a few-shot learning method (Brown et al., 2020) enables LLMs to adapt to new tasks by utilizing only a small number of context demonstration

examples. Formally, given an LLM f_θ parameterized by θ, a test query $(x_{\text{test}}, y_{\text{test}})$, and a set of k labeled demonstrations $\mathcal{D}_{\text{context}} = \{(x_i, y_i)\}_{i=1}^{k}$, the probability of generating the target label y_{test} using f_θ is formulated as:

$$p(y_{\text{test}}|c, x_{\text{test}}) = f_\theta(\mathcal{V}(y_{\text{test}})|c, \mathcal{T}(x_{\text{test}}, \cdot)), \quad (1)$$

where $\mathcal{V}(\cdot)$ is a verbalizer that maps the task labels y to words $\mathcal{V}(y)$ in the LLM's vocabulary. $\mathcal{T}(\cdot)$ denotes the process of wrapping up the input using the instruction prompt template. c is the input context formed by concatenating the k in-context demonstrations from $\mathcal{D}_{\text{context}}$, i.e., $c = \mathcal{T}(x_1, y_1) \oplus \mathcal{T}(x_2, y_2) \oplus \ldots \oplus \mathcal{T}(x_k, y_k)$ where $\oplus$ denotes concatenation.

3.2 Background on Submodularity

Submodular optimization has achieved great success in many machine learning tasks of finding diverse subsets, including text, image and video summarization (Lin and Bilmes, 2011, 2012; Gygli et al., 2015; Lavania et al., 2021; Kumari and Bilmes, 2021), feature selection (Liu et al., 2013; Zheng et al., 2014), curriculum learning (Zhou and Bilmes, 2018; Zhou et al., 2020), active learning (Guillory and Bilmes, 2011; Wei et al., 2015), training data selection (Wei et al., 2014), etc.

Submodular functions (Fujishige, 2005) are widely recognized to model notions of diversity, representativeness, and coverage in many applications (Bilmes, 2022). These functions satisfy the diminishing returns property i.e., the incremental benefit of adding a new element decreases as the context size increases. Mathematically, given a ground set V, the submodular function $f : 2^V \rightarrow \mathbb{R}$ must satisfy $f(A \cup \{v\}) - f(A) \geq f(B \cup \{v\}) - f(B)$ for subsets $A \subseteq B \subseteq V$ and any $v \in V \setminus B$. Given a submodular function f that is non-negative ($\forall A \subseteq V, f(A) \geq 0$) and monotone ($\forall A \subseteq B \subseteq V, f(A) \leq f(B)$), the ground set V can be summarized via submodular maximization under a cardinality constraint, i.e., $\max_{A \subseteq V, |A| \leq k} f(A)$. This can be approximated with a $(1 - \frac{1}{e})$ constant factor guarantee using the greedy algorithm (Nemhauser et al., 1978; Minoux, 1978) described in Appendix A. We assume all submodular functions discussed in this paper are non-negative and monotone.

4 Proposed Framework: Div-S3

Div-S3 addresses two essential questions pertinent to data and label-efficient ICL:

(1) *Given an unlabeled pool of target task samples $\mathcal{X}_{unlabeled}$, how can we identify the most informative set of examples $\mathcal{X}_{labeled}$ to annotate and use as demonstrations for the target tasks' queries?*

This question focuses on the data and label efficiency aspect of ICL using LLMs. Similar to Margatina et al. (2023), this stage of "**Exemplar Annotation**" can be viewed as one iteration of pool-based active learning, aimed at choosing a set of the most informative and diverse demonstrations/exemplars for annotation. After augmenting the samples in $\mathcal{X}_{\text{labeled}}$ with their respective labels, we denote the resultant labeled set as $\mathcal{D}_{\text{labeled}}$.

(2) *Given a query x_{test}, how can we select the most relevant and non-redundant exemplars from $\mathcal{D}_{labeled}$?*

This addresses the "**Exemplar Retrieval**" stage of ICL, where the goal is to retrieve/select the most relevant exemplars from an annotated pool of exemplars given a particular test query.

4.1 Exemplar Annotation

In this stage, we assume access to a large pool of unlabeled samples denoted by $\mathcal{D}_{\text{unlabeled}}$ belonging to the target task. Unlike the similarity-based active learning approach explored in Margatina et al. (2023), we do not assume access to the full test set during the exemplar annotation phase. Performing similarity-based sampling of exemplars for each test query individually would not be an efficient use of labeling resources, as the annotation budget could increase linearly with the number of test queries considered, overlooking the shared information among exemplars.

Given $\mathcal{X}_{\text{unlabeled}}$ of size n, this stage aims to select $\mathcal{X}_{\text{labeled}} \subseteq \mathcal{X}_{\text{unlabeled}}$ such that $|\mathcal{X}_{\text{labeled}}| = k$ and $k << n$. This process is performed only once and can be viewed as one iteration of pool-based active learning (Settles, 2011). As this stage determines the examples utilized as in-context demonstrations in the second stage, we aim to curate a set of diverse and representative samples that can comprehensively cover the target task space. To achieve this, we utilize a submodular function instantiated on the entire ground set $V = \mathcal{X}_{\text{unlabeled}} = \{x_i\}_{i=1}^{n}$.

While our framework applies to any submodular functions, for the experiments in this paper, we use a popular submodular function called *facility location* (Cornuejols et al., 1977; Mirchandani and Francis, 1990), which is closely related to but more general than k-medoid clustering. Given a similar-

ity metric $\text{sim}(\cdot,\cdot)$, the facility location function is defined as follows:

$$f(A) = \sum_{s_i \in V} \max_{s_j \in A} \text{sim}(s_i, s_j) \quad (2)$$

To obtain our $\mathcal{D}_{\text{labeled}}$ set, we first use a pre-trained language model g_θ (sentence-BERT (Reimers and Gurevych, 2019) in this work) to compute text embeddings of each sample in the unlabeled pool such that s_i in Eq. 2 is $g_\theta(x_i)\ \forall\ x_i \in \mathcal{X}_{\text{unlabeled}}$. Using the text embeddings of samples in $\mathcal{X}_{\text{unlabeled}}$, we compute a similarity matrix using a tuned similarity kernel (further discussed in Sec. C), use it to instantiate the facility location function, and then maximize the submodular objective to obtain our final set of demonstrations for annotation as follows:

$$\mathcal{X}_{\text{labeled}} \in \text{argmax}_{A \subseteq \mathcal{X}_{\text{unlabeled}}, |A| \leq k} f(A) \quad (3)$$

4.2 Exemplar Retrieval

After obtaining a set $\mathcal{D}_{\text{labeled}}$ of diverse and representative annotated exemplars, the next step is to select exemplars from $\mathcal{D}_{\text{labeled}}$ given a test query x_{test}. Previous studies in ICL (Rubin et al., 2021; Liu et al., 2022; Su et al., 2022; Margatina et al., 2023) have employed cosine similarity-based ranking to select exemplars that are most similar to the test query. However, when each example's relevance (or similarity) to the test query is considered independently, it may yield relevant but similar exemplars, carrying redundant information that is wasteful for inference.

To reduce the redundancy amongst the selected in-context exemplars, we formalize the exemplar retrieval stage as a conditional submodular subset selection problem. We use the same submodular function utilized in the prior stage for exemplar annotation, but unlike the previous stage, which employs a generic summarization approach as shown in Eq. 3, this stage focuses on conducting query-based (or conditional) summarization, aiming to summarize the labeled set $\mathcal{X}_{\text{labeled}}$ given a query set containing x_{test}. When performing ICL using LLMs, the query x_{test} is the test input instance.

In this work, we utilize a two-phase method named **Submodular Span Summarization (S3)** (Kumari and Bilmes, 2021) to perform query-focused summarization of an annotated pool given a test query. Our objective is to obtain a set of exemplars that are not only relevant to the test query but also encompass diverse aspects crucial for aiding the LLM in the target task.

Phase 1 of S3 targets selecting a relatively large subset relevant to the query set. Mathematically, given a ground set V that includes the query set Q and the data being summarized V_Q (where $V_Q = V \setminus Q$), and a submodular function f defined on the entire ground set V, the submodular span optimization problem is defined as follows:

$$\max_{A \subseteq V_Q} |A| \quad \text{s.t. } f(A|Q) \leq \epsilon \quad (4)$$

where $\epsilon \geq 0$ is a small scalar controlling the desired relevance level. $f(A|Q) := f(A \cup Q) - f(Q)$ denotes the conditional gain of set A given the query set Q. Low $f(A|Q)$ represents high conditional redundancy of A given query set Q. Thus, to optimize Eq. 4, we get a set A with a low Q-conditioned f-valuation. The dual to this problem is shown below, where we minimize the conditional gain $f(A|Q)$ subject to a lower-bound cardinality constraint:

$$\min_{A \subseteq V_Q} f(A|Q) \quad \text{s.t. } |A| \geq k_1 \quad (5)$$

That is, the above optimization problem is cardinality-constrained submodular minimization, which does not have a constant factor approximation algorithm (Svitkina and Fleischer, 2011). Similar to Kumari and Bilmes (2021), we utilize a modular approximation of $f(A|Q)$, i.e., $m_Q(A) = \sum_{a \in A} f(a|Q)$ to optimize Eq. 5. Note that $m_Q(A) \geq f(A|Q)$ is an upper bound of $f(A|Q)$. Theoretical guarantees based on the curvature of the submodular functions can be found in Kumari and Bilmes (2021).

For the Exemplar Retrieval stage of ICL, the data to be summarized V_Q is the annotated pool of exemplars $\mathcal{X}_{\text{labeled}}$. We denote our solution of S3's Phase 1 as A_Q, essentially the annotated exemplars relevant to the input query Q.

Phase 2 of S3 focuses on diverse aspects of various relevant exemplars, ensuring the final set of exemplars for ICL is both relevant and non-redundant. We summarize the S3 Phase 1 resultant set A_Q by performing submodular maximization subject to a cardinality (or a knapsack) constraint (Eq. 6). In this paper's experiments, we use the same submodular function for the Exemplar Annotation stage and the two phases of S3 for the Exemplar retrieval stage. Our framework is

general, and different submodular functions can apply if various diversity properties are desirable.

$$\max_{A \subseteq A_Q, |A| \leq k_2} f(A) \tag{6}$$

Since the final set of exemplars selected will be used as in-context demonstrations, we can also apply a knapsack constraint for the constraint associated with Eq. 6.

$$\max_{A \subseteq A_Q} f(A)$$
$$\text{s.t.} \sum_{a \in A} c(a) \leq b \tag{7}$$

The budget b can be set to the difference between the pre-set context window of the inference LLM and the token length of the formatted test query. The cost $c(a)$ denotes the cost associated with exemplar a that can be set to the token length of the instruction formatted exemplar a, i.e., $\mathcal{T}(x_a, y_a)$. Note that the knapsack constraint generalizes the cardinality constraint: for cardinality constraints, $c(a)$ is simply 1, and the budget is k_2.

To optimize Eq. 7, we use the modified greedy algorithm presented in Lin and Bilmes (2010) with a $(1 - 1/\sqrt{e})$ constant factor approximation factor under certain conditions. The modified greedy at each iteration i selects exemplar s_i with the largest ratio of objective conditional gain to the scaled cost, i.e., $s_i = \text{argmax}_{s \in A_Q} \frac{f(A_{i-1} \cup s) - f(A_{i-1})}{c(s)^r}$ if $\sum_{s \in A_{i-1} \cup s_i} c(s) \leq b$.

We provide a detailed analysis of the computational complexity associated with the two stages of Div-S3 in Appendix B. Since both stages discussed in Sec. 4.1 and 4.2 involve certain hyperparameters such as the similarity kernel, kernel width (in case of using the RBF kernel), the budget of Phase 1 of S3 (k_1), budget associated with Phase 2 of S3 (k_2 when using a cardinality constraint), or the scaling factor (r when using a knapsack constraint), we utilize a separate validation set for hyperparameter tuning(Appendix C). Our approach, thus, needs minimal supervision and is learning-free as it does not involve fine-tuning the inference LLM on any task-specific task.

5 Experiments

To demonstrate the effectiveness of our proposed framework for exemplar annotation and retrieval for performing ICL, we conduct experiments over a diverse set of 7 NLP datasets using a suite of five different LLMs as in-context learners. Since our proposed framework *Div-S3* does not perform any task-specific LLM fine-tuning, we only compare it to existing learning-free ICL methods to ensure a fair comparison.

5.1 Datasets

We evaluate *Div-S3* on the following 7 NLP datasets, which cover five distinct tasks:

SST-5 (Socher et al., 2013): This dataset involves sentiment classification of movie reviews into five distinct sentiment categories: very negative, negative, neutral, positive, and very positive.

SST-2 (Socher et al., 2013): Similar to SST-5, this dataset also deals with sentiment classification into positive and negative labels.

RTE (Bentivogli et al., 2009): The Recognizing Textual Entailment dataset focuses on discerning textual entailment and belongs to a broader scope of tasks studied under Natural Language Inference. Given a pair of sentences, the goal is to determine whether the premise (also called text) entails (or implies) the hypothesis.

MRPC (Dolan and Brockett, 2005): This dataset deals with paraphrase detection task where given a pair of sentences, the objective is to determine whether they are semantically equivalent.

TREC (Li and Roth, 2002): This dataset involves a question classification task. In this work, we focus only on the six coarse labels: Abbreviation, Entity, Description, Human being, Location, and Numeric value.

DBpedia (Lehmann et al., 2015): Here, we have a topic classification dataset constructed by selecting 14 non-overlapping classes from the base DBpedia 2014. The 14 different classes are as follows: company, educational institution, artist, athlete, office holder, means of transportation, building, natural place, village, animal, plant, album, film, and written work.

HellaSwag (Zellers et al., 2019): In this case, we have another NLI dataset studying grounded commonsense reasoning. It consists of multiple-choice questions with four answer choices. Three out of four choices are incorrect and designed in an adversarial way to deceive machines without misleading humans.

We provide the prompt templates used for different datasets in Appendix E and dataset split statistics in Appendix F.

Method	SST-5	SST-2	RTE	TREC	MRPC	HellaSwag	DBPedia	Average
Random-Similar	47.18	88.15	57.40	71.00	63.32	66.09	90.22	69.05
VoteK-Similar	44.30	89.19	52.43	71.09	63.91	**66.48**	89.51	68.13
Div-Similar	45.02	90.48	53.07	77.60	67.89	66.14	90.94	70.16
Div-MixModSub	45.93	90.25	54.87	76.80	67.89	66.18	**91.06**	70.43
Random-S3	46.73	88.23	58.24	70.80	63.32	66.18	89.49	69.00
Div-S3 (cardinality)*	49.59	**91.28**	**60.65**	80.00	**68.63**	66.32	89.87	**72.33**
Div-S3 (knapsack)*	**49.73**	-	59.57	**80.60**	67.40	66.44	90.69	72.24

Table 1: ICL performance on different NLP datasets using GPT-J-6B as our inference LLM. Here, we compare *Div-S3* against the baselines described in Sec. 5.3. Our proposed methods are marked with an asterisk (*).

Method	SST-5	SST2	RTE	TREC
Random-Similar	41.01	70.26	53.91	62.60
VoteK-Similar	39.97	64.71	54.17	62.11
Div-S3 (cardinality)*	40.36	**77.64**	53.79	66.40
Div-S3 (knapsack)*	**41.49**	-	**54.87**	**68.80**

(a) GPT-Neo 2.7B

Method	SST-5	SST-2	RTE	TREC
Random-Similar	35.78	78.63	48.26	55.00
VoteK-Similar	**39.06**	80.34	48.70	52.99
Div-S3 (cardinality)*	37.29	**82.80**	**52.71**	62.80
Div-S3 (knapsack)*	38.51	-	48.74	**65.00**

(b) GPT-Neo 1.3 B

Table 2: ICL performance on four candidate datasets using two *GPT-Neo* models. Our proposed methods are marked with an asterisk (*).

5.2 Models

We evaluated our proposed framework and compared baseline methods for ICL on five LLMs of varying complexity belonging to the GPT (Radford et al., 2019; Brown et al., 2020) and OPT (Zhang et al., 2022a) families. Specifically, the largest model that we used as the inference LLM is GPT-J-6B (Wang and Komatsuzaki, 2021), and from both GPT-Neo (Black et al., 2021) and OPT families, we used their respective 2.7B and 1.3B variants.

5.3 Baselines

We compare *Div-S3* to the learning-free baselines listed below. In all listed methods, the first part of the name preceding the hyphen indicates the strategy used for exemplar annotation (stage 1) while the later part denotes the strategy used for exemplar retrieval (stage 2). The second stage method named "*Similar*" uses cosine similarity-based ranking to extract the most relevant exemplars at test time.

Random-Similar: randomly selects exemplars from the unlabeled pool for annotation and then uses Similar for exemplar retrieval.

Random-S3: randomly selects exemplars during the annotation phase and then uses S3 to select relevant exmplars during the retrieval phase.

VoteK-Similar (Su et al., 2022): uses graph-based method named VoteK to select $k/10$ samples which are diverse in the feature space used. For selecting the remaining $9k/10$ samples, the inference LLM is used to compute the average log probability over the generated output to select diverse exemplars in terms of confidence scores. For the retrieval stage, Similar is used.

Div-Similar (Cornuejols et al., 1977; Mirchandani and Francis, 1990; Balakrishnan et al., 2022): maximizes a submodular facility location objective to select a subset of Diverse and representative samples for annotation. This strategy differs from *Div-S3* where we use Submodular Span Summarization for exemplar retrieval instead of Similar.

Div-MixModSub: uses a facility location-based submodular maximization to select Diverse exemplars for annotation. Similar to Kumari and Bilmes (2021), we consider another baseline named MixModSub, which uses a mixture of a submodular function (facility location) and a modular function (similarity scores) to jointly balance representativeness and query-relevance in the final retrieved set.

We report average performance across three runs for the following baselines: Random-Similar, Random-S3, and VoteK-Similar. Table 8 in the appendix presents the standard deviations for these baselines.

6 Results

We compare *Div-S3* to the baselines discussed in Sec 5.3 in Tables 1, 2, and 3 using GPT-J-6B, GPT-Neo, and OPT models respectively. We fix the

Method	SST-5	SST2	RTE	TREC
Random-Similar	37.29	74.27	51.38	58.27
VoteK-Similar	36.72	73.96	**53.52**	50.39
Div-S3 (cardinality)*	37.74	**79.82**	53.43	64.40
Div-S3 (knapsack)*	**38.69**	-	51.26	**65.60**

(a) OPT 2.7B

Method	SST-5	SST-2	RTE	TREC
Random-Similar	**32.64**	81.69	52.47	63.53
VoteK-Similar	28.52	81.38	52.73	61.07
Div-S3 (cardinality)*	32.08	**86.12**	53.79	**67.60**
Div-S3 (knapsack)*	31.54	-	**54.87**	67.20

(b) OPT 1.3 B

Table 3: ICL performance on four datasets using two *OPT* models.

Method	SST-5	SST-2	RTE	TREC
Similar	50.95	89.91	54.15	**90.80**
S3 (cardinality)*	50.81	**91.28**	55.96	89.00
S3 (knapsack)*	**52.08**	-	**57.04**	90.00

Table 4: ICL performance on four candidate datasets using GPT-J-6B model in the non-active learning setting where only Stage 2 focusing on exemplar retrieval is active.

annotation budget as 100 i.e., $|\mathcal{D}_{\text{labeled}}| = 100$ for all tasks and methods. For Div-S3 when using a knapsack constraint, the budget (in terms of the token length) is the inference LLM's pre-set context window size minus the formatted test query length. The pre-set context window size is 1,024 for the models we study in this work.

On the SST-5 and RTE datasets, *Div-S3* demonstrates roughly 3% absolute gain compared to the baseline models in terms of accuracy. Across all tasks, we see that *Div-Similar* is a strong baseline that outperforms the other two baselines: *Random-Similar* and *VoteK-Similar*. This indicates that selecting a diverse and representative set of exemplars during the annotation stage is crucial to the overall performance of ICL. Integrating *Div* (Stage 1) with *S3* (Stage 2) results in additional improvements, affirming our hypothesis that the final stage of exemplar retrieval at test time should be treated as a subset selection problem rather than independently selecting exemplars using modular similarity-based values.

When using the smaller LLMs belonging to the GPT-Neo and OPT families on four candidate datasets, we observe that *Div-S3* outperforms other baselines. Notably, across SST-2 and TREC datasets, it achieves a maximum absolute gain of approximately 10% in terms of accuracy. This demonstrates that our proposed framework *Div-S3* exhibits consistent improvements across various tasks and LLM variants.

6.1 Sensitivity Analysis

Non-Active Learning setting: In this section, we verify the effectiveness of the *S3* method for exemplar retrieval in a non-active learning setting where there are no constraints on the annotation budget. Here we consider the entire training set as the annotated pool of exemplars, meaning $|\mathcal{D}_{\text{labeled}}| = n$. This renders the *Exemplar Annotation* stage redundant. In Table 4 and 5 when using GPT-J-6B and OPT models resp., we compare *S3* (when using cardinality and knapsack constraints for phase 2 of S3) to *Similar*. As depicted in Table 4 and 5, *S3* consistently outperforms the modular selection method *Similar*, reinforcing our hypothesis of approaching exemplar retrieval as a subset selection problem.

Order Sensitivity: Prior works such as Zhao et al. (2021) and Lu et al. (2022) have shown that ICL's performance is extremely sensitive to the ordering of the exemplars in the input prompt with performances varying between random-guess levels to fine-tuning based state-of-the-art levels. In this section, our goal is to demonstrate how *Div-S3* is less sensitive to the ordering of the retrieved exemplars in the LLM's input prompt. To do this, we fix the strategy used during the exemplar annotation stage as *Div* and select exemplars from the annotated pool of exemplars using three different methods: *Random, Similar*, and *S3*. Given m selected exemplars, there are overall m possible order permutations. Scoring all m orderings by making inference calls to the LLM would be computationally challenging, so we randomly sample 50 different orderings and score those across different *Stage 2* methods. In Figure 3, we demonstrate the performance variation corresponding to these random orderings on three candidate datasets, and it can be seen that *Div-S3* achieves better average accuracy while being less sensitive to the order of the exemplars compared to other baselines due to its ability to balance relevance and diversity during the exemplar retrieval stage.

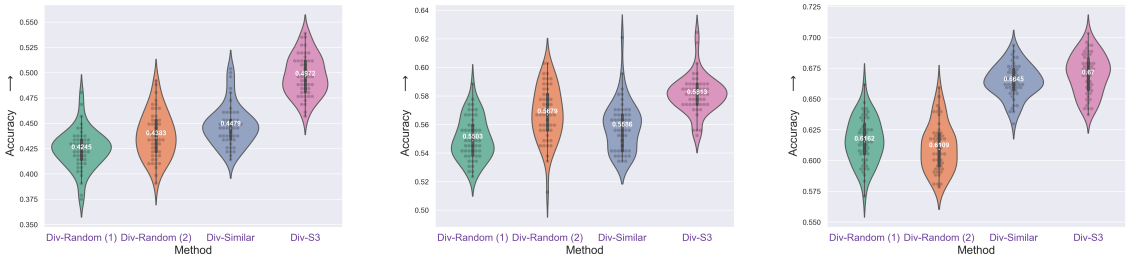
Method	SST-5	SST2	RTE	TREC
Similar	45.25	83.94	48.01	79.60
S3 (cardinality)*	44.34	**85.78**	**53.07**	**80.80**
S3 (knapsack)*	**45.97**	-	50.18	**80.80**

(a) OPT 2.7B

Method	SST-5	SST-2	RTE	TREC
Similar	36.06	87.50	51.99	77.60
S3 (cardinality)*	37.92	**88.30**	**53.07**	78.20
S3 (knapsack)*	**38.55**	-	51.26	**79.00**

(b) OPT 1.3 B

Table 5: ICL performance on four candidate datasets using two *OPT* models in the non-active learning setting where only Stage 2 focusing on exemplar retrieval is active. Our proposed methods are marked with an asterisk (*).



(a) SST-5 dataset (b) RTE dataset (c) MRPC dataset

Figure 3: Sensitivity analysis of the set of exemplars retrieved by different *Exemplar Retrieval* methods to their ordering in the LLM's input prompt. The first stage of Exemplar annotation is identical in all four methods studied. When using *Random* as the second stage method, we report the sensitivity results for two different random seeds.

7 Conclusion

We propose *Div-S3*, which unites the two stages of exemplar selection using submodular optimization. For the Exemplar Annotation stage, we select a diverse subset of exemplars for annotation under a fixed budget utilizing submodular maximization under a cardinality constraint. For the Exemplar Retrieval stage, we utilize a Submodular Span Summarization approach that finds diverse annotated exemplars that are most relevant to the test instance. Compared to previous methods on the exemplar selection task, *Div-S3* models both diversity and relevance for the second stage, and incorporates the rich class of submodular functions. On multiple NLP tasks and using various LLMs, *Div-S3* shows consistent improvements. *Div-S3* is also more robust to the ordering of the exemplars empirically as we account for diversity in the exemplar retrieval stage.

8 Ethical Discussion

As a method to facilitate ICL, *Div-S3* is prone to unethical exemplars and harmful annotations. With unethical exemplars collected in the initial unlabeled pool of exemplars, *Div-S3* could select some of them for annotation and use them as context for inference, which may result in unethical generations from the LLM. Moreover, we assume the annotation process is trustworthy, but if biases are present in the labels, the resulting selection process and the inference results could still contain biases. We will research solving the potential ethical issues in future work.

9 Limitations & Future Work

In this study, the largest inference LLM that we use for ICL has 6B parameters. It would be interesting to see how the performance of *Div-S3* transfers to more heavyweight LLMs such as LLaMA-13B, 70B (Touvron et al., 2023), PaLM (Chowdhery et al., 2022), GPT-3.5, 4 (Brown et al., 2020; OpenAI, 2023), etc. The submodular function used during both stages of Div-S3 is a facility location function. Since our framework is general, one can plug in any submodular function such as graph cut function, feature-based submodular function, etc. Defining a mixture of submodular functions to control more fine-grained aspects of relevance along with diversity could be an interesting research direction.

Acknowledgements

We thank all reviewers and the area chairs for their valuable feedback. This work was supported in part by the CONIX Research Center, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA. This material is also based upon work supported by the National Science Foundation RINGs program under Grant No. 2148367.

References

Ravikumar Balakrishnan, Tian Li, Tianyi Zhou, Nageen Himayat, Virginia Smith, and Jeff Bilmes. 2022. Diverse client selection for federated learning via submodular maximization. In *International Conference on Learning Representations*.

Luisa Bentivogli, Bernardo Magnini, Ido Dagan, Hoa Trang Dang, and Danilo Giampiccolo. 2009. The fifth PASCAL recognizing textual entailment challenge. In *Proceedings of the Second Text Analysis Conference, TAC 2009, Gaithersburg, Maryland, USA, November 16-17, 2009*. NIST.

Jeff Bilmes. 2022. Submodularity in machine learning and artificial intelligence. *arXiv preprint arXiv:2202.00132*.

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. Gpt-neo: Large scale autoregressive language modeling with mesh-tensorflow. *If you use this software, please cite it using these metadata*, 58:2.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

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.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Stephanie Chan, Adam Santoro, Andrew Lampinen, Jane Wang, Aaditya Singh, Pierre Richemond, James McClelland, and Felix Hill. 2022. Data distributional properties drive emergent in-context learning in transformers. *Advances in Neural Information Processing Systems*, 35:18878–18891.

Ting-Yun Chang and Robin Jia. 2023. Data curation alone can stabilize in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8123–8144.

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*.

Gerard Cornuejols, Marshall Fisher, and George L Nemhauser. 1977. On the uncapacitated location problem. In *Annals of Discrete Mathematics*, volume 1, pages 163–177. Elsevier.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. 2023. Why can gpt learn in-context? language models secretly perform gradient descent as meta-optimizers. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4005–4019.

Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric Xing, and Zhiting Hu. 2022. Rlprompt: Optimizing discrete text prompts with reinforcement learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3369–3391.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*.

Satoru Fujishige. 2005. *Submodular functions and optimization*. Elsevier.

Andrew Guillory and Jeff A Bilmes. 2011. Online submodular set cover, ranking, and repeated active learning. *Advances in neural information processing systems*, 24.

Michael Gygli, Helmut Grabner, and Luc Van Gool. 2015. Video summarization by learning submodular mixtures of objectives. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3090–3098.

Chi Han, Ziqi Wang, Han Zhao, and Heng Ji. 2023. In-context learning of large language models explained as kernel regression. *arXiv preprint arXiv:2305.12766*.

Joel Jang, Seonghyeon Ye, and Minjoon Seo. 2023. Can large language models truly understand prompts? a case study with negated prompts. In *Transfer Learning for Natural Language Processing Workshop*, pages 52–62. PMLR.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Abdullatif Köksal, Timo Schick, and Hinrich Schütze. 2022. Meal: Stable and active learning for few-shot prompting. *arXiv preprint arXiv:2211.08358*.

Jannik Kossen, Tom Rainforth, and Yarin Gal. 2023. In-context learning in large language models learns label relationships but is not conventional learning. *arXiv preprint arXiv:2307.12375*.

Lilly Kumari and Jeff Bilmes. 2021. Submodular span, with applications to conditional data summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12344–12352.

Po-Nien Kung and Nanyun Peng. 2023. Do models really learn to follow instructions? an empirical study of instruction tuning. *arXiv preprint arXiv:2305.11383*.

Chandrashekhar Lavania, Kai Wei, Rishabh Iyer, and Jeff Bilmes. 2021. A practical online framework for extracting running video summaries under a fixed memory budget. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, pages 226–234. SIAM.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195.

Xiaonan Li and Xipeng Qiu. 2023. Finding supporting examples for in-context learning. *arXiv preprint arXiv:2302.13539*.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Yingcong Li, Muhammed Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak. 2023. Transformers as algorithms: Generalization and stability in in-context learning. In *International Conference on Machine Learning*, pages 19565–19594. PMLR.

Hui Lin and Jeff Bilmes. 2010. Multi-document summarization via budgeted maximization of submodular functions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 912–920.

Hui Lin and Jeff Bilmes. 2011. A class of submodular functions for document summarization. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 510–520.

Hui Lin and Jeff Bilmes. 2012. Learning mixtures of submodular shells with application to document summarization. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, pages 479–490.

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

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*.

Yuzong Liu, Kai Wei, Katrin Kirchhoff, Yisong Song, and Jeff Bilmes. 2013. Submodular feature selection for high-dimensional acoustic score spaces. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7184–7188. IEEE.

Sheng Lu, Irina Bigoulaeva, Rachneet Sachdeva, Harish Tayyar Madabushi, and Iryna Gurevych. 2023. Are emergent abilities in large language models just in-context learning? *arXiv preprint arXiv:2309.01809*.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098.

Katerina Margatina, Timo Schick, Nikolaos Aletras, and Jane Dwivedi-Yu. 2023. Active learning principles for in-context learning with large language models. *arXiv preprint arXiv:2305.14264*.

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.

Michel Minoux. 1978. Accelerated greedy algorithms for maximizing submodular set functions. In *Optimization Techniques: Proceedings of the 8th IFIP Conference on Optimization Techniques Würzburg, September 5–9, 1977*, pages 234–243. Springer.

Pitu B Mirchandani and Richard L Francis. 1990. *Discrete location theory*.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. 1978. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294.

OpenAI. 2023. Gpt-4 technical report.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2021. Learning to retrieve prompts for in-context learning. *arXiv preprint arXiv:2112.08633*.

Burr Settles. 2011. From theories to queries: Active learning in practice. In *Active learning and experimental design workshop in conjunction with AISTATS 2010*, pages 1–18. JMLR Workshop and Conference Proceedings.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Taylor Sorensen, Joshua Robinson, Christopher Rytting, Alexander Shaw, Kyle Rogers, Alexia Delorey, Mahmoud Khalil, Nancy Fulda, and David Wingate. 2022. An information-theoretic approach to prompt engineering without ground truth labels. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 819–862.

Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, et al. 2022. Selective annotation makes language models better few-shot learners. *arXiv preprint arXiv:2209.01975*.

Zoya Svitkina and Lisa Fleischer. 2011. Submodular approximation: Sampling-based algorithms and lower bounds. *SIAM Journal on Computing*, 40(6):1715–1737.

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*.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. 2023. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pages 35151–35174. PMLR.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

Albert Webson and Ellie Pavlick. 2022. Do prompt-based models really understand the meaning of their prompts? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2300–2344, Seattle, United States. Association for Computational Linguistics.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, et al. 2023. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*.

Kai Wei, Rishabh Iyer, and Jeff Bilmes. 2015. Submodularity in data subset selection and active learning. In *International conference on machine learning*, pages 1954–1963. PMLR.

Kai Wei, Yuzong Liu, Katrin Kirchhoff, Chris Bartels, and Jeff Bilmes. 2014. Submodular subset selection for large-scale speech training data. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3311–3315. IEEE.

Zhiyong Wu, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. 2022. Self-adaptive in-context learning. *arXiv preprint arXiv:2212.10375*.

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

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations*.

Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. 2023. Compositional exemplars for in-context learning. *arXiv preprint arXiv:2302.05698*.

Fan Yin, Jesse Vig, Philippe Laban, Shafiq Joty, Caiming Xiong, and Chien-Sheng Jason Wu. 2023. Did you read the instructions? rethinking the effectiveness of task definitions in instruction learning. *arXiv preprint arXiv:2306.01150*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022a. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Yiming Zhang, Shi Feng, and Chenhao Tan. 2022b. Active example selection for in-context learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9134–9148.

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.

Jingjing Zheng, Zhuolin Jiang, Rama Chellappa, and Jonathon P Phillips. 2014. Submodular attribute selection for action recognition in video. *Advances in Neural Information Processing Systems*, 27.

Tianyi Zhou and Jeff Bilmes. 2018. Minimax curriculum learning: Machine teaching with desirable difficulties and scheduled diversity. In *International Conference on Learning Representations*.

Tianyi Zhou, Shengjie Wang, and Jeffrey Bilmes. 2020. Curriculum learning by dynamic instance hardness. *Advances in Neural Information Processing Systems*, 33:8602–8613.

A Maximizing monotone submodular functions

In this section, we provide the outline of the greedy algorithm (Nemhauser et al., 1978) used for maximizing monotone submodular functions subject to a cardinality constraint (Eqs. 3 and 6).

Algorithm 1 Greedy Submodular Maximization (Nemhauser et al., 1978)

Input: Polymatroid function $f : 2^V \rightarrow \mathbb{R}$, cardinality constraint k
Output: Set $A \subseteq V$ maximizing $f(A)$ under cardinality constraint k
Initialize an empty set $A \leftarrow \emptyset$
for $j = 1$ to k **do**
  $e \leftarrow \text{argmax}_{v \in V \setminus A}(f(A \cup \{v\}) - f(A))$
  $A \leftarrow A \cup \{e\}$
end for
return S

B Complexity Analysis

Exemplar Annotation To select the set of exemplars to label, we are required to maximize the facility location function using greedy selection. This step requires 1) constructing a pairwise similarity matrix and 2) applying greedy selection. Given $|\mathcal{X}_{\text{unlabeled}}| = n$ and $|\mathcal{X}_{\text{labeled}}| = k$, constructing the pairwise similarity matrix requires $\mathcal{O}(n^2)$ operations while greedy selection requires $\mathcal{O}(nk)$, yielding a final complexity of $\mathcal{O}(n^2 + nk)$. In practice, the quadratic cost of computing the similarity matrix can be reduced by utilizing sparse matrices, while greedy selection can be sped up significantly by using a priority queue (Minoux, 1978). The exemplar annotation cost is incurred only once for each task.

Exemplar Retrieval At inference time, we retrieve exemplars annotated in the first stage that are relevant to a particular query set by using S3. The first phase of S3 requires minimizing a modular function $m_Q(A)$, to select k_1 query-relevant samples from a set of k labeled exemplars. Thus, this phase has a time complexity of $\mathcal{O}(k + k \log k_1)$.

The second phase of S3 summarizes the k_1 samples down to a diverse set of k_2 samples by maximizing a facility location function subject to a knapsack constraint. Similar to the process of examplar annotation, this phase also requires constructing a pairwise similarity matrix ($\mathcal{O}(k_1^2)$, though this can simply be reused from the exemplar annotation phase as we do not alter the similarity metric. Thus, the complexity of this phase is $\mathcal{O}(k_1 k_2)$ which is incurred by performing greedy selection.

In our experiments, we focus on the low-data regime and set the annotation budget, i.e., k, as 100 in the active-learning setting. As can be seen in Sec. C, $k1$ is often less than 30, making the entire process utilizing $S3$ for exemplar retrieval very efficient and significantly faster than the LLM inference call by several orders of magnitude.

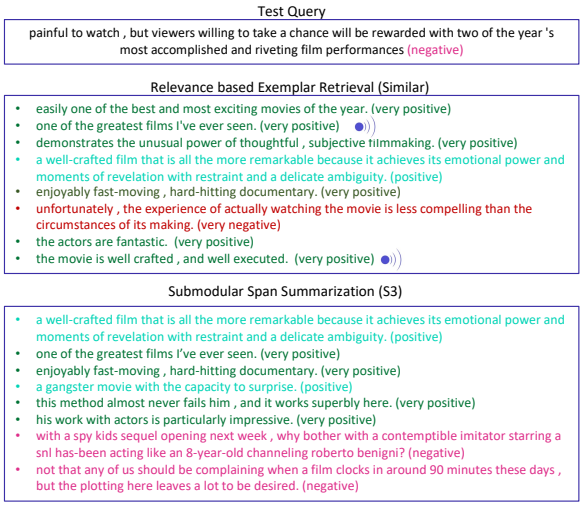


Figure 4: Another sample test query from the SST-5 dataset with its corresponding set of exemplars selected using *Similar* and *S3* from a limited exemplars pool. Exemplars are colored based on their class names. We use echo lines to denote the redundant exemplars chosen by *Similar* and used as a part of the input context during ICL.

C Hyperparameter tuning & Compute resources

Div-S3 involves a set of hyperparameters listed at the end of Sec. 4.1. To tune these hyperparameters, we use the validation set of each dataset. In case the validation set is unavailable, we randomly select 10% of samples from the training set while using the remaining 90% as the unlabeled pool of exemplars. Specifically, for the similarity metric needed to instantiate the facility location function, we compare the following kernels: cosine similarity with negative entries truncated to zero, modified cosine similarity adding a constant 1 to each entry, and RBF kernel with kernel widths $\sigma \in \{0.1, 0.2, 0.5, 1.0, 2.0, 5.0, 10.0\}$. While the hyperparameter search space for the kernel is quite broad, we prune the set of candidate kernels by inspecting the gains of the submodular function when performing greedy maximization as shown in Fig-

Test Query

a working class `` us vs. them '' opera that leaves no heartstring untugged and no liberal cause unplundered. (great)

Relevance based Exemplar Retrieval (Similar)

- a depressing story that throws a bunch of hot-button items in the viewer 's face and asks to be seen as hip , winking social commentary. (terrible)
- a powerful and telling story that examines (great)
- an interesting slice of history. (great)
- a riveting documentary. (great)
- of enthralling drama (great)
- with a lighthearted glow , some impudent snickers , and a glorious dose of humankind 's liberating ability (great)
- a heartfelt story (great)
- is a discreet moan of despair about entrapment in the maze of modern life (terrible)

Submodular Span Summarization (S3)

- a powerful and telling story that examines (great)
- a wonderful ensemble cast (great)
- easily one of the best and most exciting movies of the year (great)
- appears to have been made by people to whom the idea of narrative logic or cohesion is an entirely foreign concept (terrible)
- offers copious hints along the way -- myriad signs , if you will -- that beneath the familiar , funny surface is a far bigger , far more meaningful story than one in which little green men come to earth for harvesting purposes. (great)
- aims to present an unflinching look at one man 's downfall , brought about by his lack of self-awareness (great)
- makes it worth watching (great)
- a love affair (great)

Figure 5: A sample test query from the SST-2 dataset with its corresponding set of exemplars selected using *Similar* and *S3* from a limited exemplars pool. Exemplars are colored based on their class names. We use echo lines to denote the redundant exemplars chosen by *Similar* and used as a part of the input context during ICL.

ure 6. Specifically, if a kernel configuration results in the gains diminishing prematurely, then samples chosen towards the later stages of the optimization procedure may be selected randomly. Pruning such kernel configurations is relatively inexpensive, since we do not need to run the end-to-end pipeline to assess their utilities on the downstream task.

The annotation budget (k) for stage 1 is set as 100 for each task and across all LLMs studied. The budget for phase 2 of S3 (k_2) when using a cardinality constraint is roughly determined based on the average number of exemplars l that are selected by *Similar* method on each task. We search for k_2 by using neighboring values of l. To tune the budget (k_1) associated with phase 1 of Stage 2, we search over $k_1 \in \{15, 20, 25, 30, 35, 40, 45, 50\}$. For the scaling factor r when optimizing Phase 2 under a knapsack constraint, we search over $r \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$. Next, we list the best-found hyperparameters for each task in the given format (similarity kernel, k_1, r): SST-5 - (1+cosine, $40, 0.1$), RTE - (1+cosine, $30, 0.1$), MRPC - (RBF kernel with width 2.0, $15, 0.1$), SST-2 - (RBF kernel with width 5.0, 30, -), TREC - (1+cosine, $30, 0.1$), DBPedia - (1+cosine, $25, 0.4$), HellaSwag - (1+cosine, $25, 0.1$)

For all experiments, we use an A100 80GB GPU.

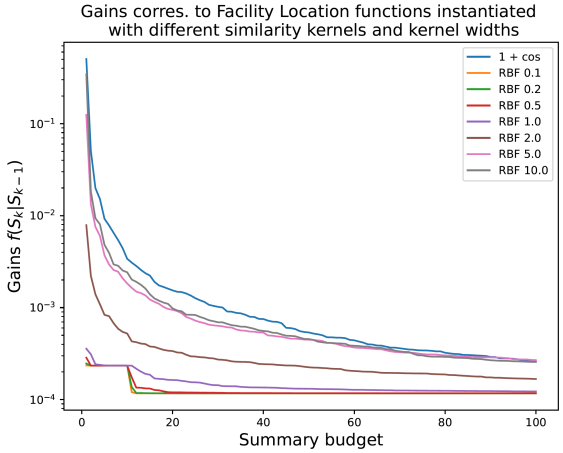


Figure 6: Plot displaying the Submodular Gains on the SST-5 dataset for different similarity kernel configurations. All kernels that use the RBF configuration with kernel width less than 1.0 result in gains that saturate (approach 0) prematurely. Such configurations are not useful for the exemplar annotation task, so they can be discarded by inspection.

D Sample examples of exemplars selected by Div-S3

In this section, we present qualitative results in the form of exemplars obtained by the following Stage 2 strategies: *Similar* and *S3*, demonstrating the effectiveness of optimizing for both diversity and query relevance when retrieving exemplars for ICL. Figure 4 shows exemplars retrieved for a particular test query belonging to the SST-5 dataset (Socher et al., 2013). Figure 5 shows exemplars retrieved for a particular test query belonging to the SST-2 dataset (Socher et al., 2013).

Dataset	Size of train set	Size of validation set	Size of test set
SST-5	8544	1101	2210
SST-2	67349	872	-
RTE	2490	277	-
MRPC	3668	408	-
TREC	5452	-	500
DBPedia	560000	-	70000
HellaSwag	39905	10042	-

Table 6: Datasets Statistics

E Prompt template

For the majority of tasks, we use the same prompt templates as Su et al. (2022). In Table 7, we provide the template used for each task explored in this work.

F Dataset Statistics

In Table 6, we provide dataset statistics of 7 datasets described in Sec. 5.1 in terms of the size of the train/validation/test splits.

Dataset	Prompt Template	Class names (Verbalizer)
SST-5	How do you feel about the following sentence? <X> answer:	very negative, negative, neutral, positive, very positive
SST-2	<X> It was	terrible, great
RTE	<X> question: <Y>. True or False? answer:	true, false
MRPC	Are the following two sentences 'equivalent' or 'not equivalent'? <X> <Y> answer:	not equivalent, equivalent
TREC	Categories: Description, Entity, Abbreviation, Human, Numeric, Location What category best describes: <X> Answer:	description, entity, abbreviation, human, numeric, location
DBpedia	title: <X>; content: <Y>	company, educational institution, artist, athlete, office holder, mean of transportation, building, natural place, village, animal, plant, album, film, written work
HellaSwag	The topic is <X>. <Y> <Z>	4 answer choices provided

Table 7: Templates of different tasks. Text marked in blue denotes the manual instruction template. Depending on the task, placeholders <X>, <Y>, and <Z> will be replaced by their available components.

Method	SST-5	SST-2	RTE	TREC	MRPC	HellaSwag	DBPedia
Random-Similar	$47.18_{1.94}$	$88.15_{0.62}$	$57.40_{1.56}$	$71.00_{3.05}$	$63.32_{1.68}$	$66.09_{0.15}$	$90.22_{1.12}$
VoteK-Similar	$44.30_{3.14}$	$89.19_{3.80}$	$52.43_{3.65}$	$71.09_{0.84}$	$63.91_{3.83}$	$\mathbf{66.48}_{2.64}$	$89.51_{2.40}$
Div-Similar	45.02	90.48	53.07	77.60	67.89	66.14	90.94
Div-MixModSub	45.93	90.25	54.87	76.80	67.89	66.18	**91.06**
Random-S3	$46.73_{0.70}$	$88.23_{1.67}$	$58.24_{3.36}$	$70.80_{1.83}$	$63.32_{2.62}$	$66.18_{0.29}$	$89.49_{0.59}$
Div-S3 (cardinality)*	49.59	**91.28**	**60.65**	80.00	**68.63**	66.32	89.87
Div-S3 (knapsack)*	**49.73**	-	59.57	**80.60**	67.40	66.44	90.69

Table 8: ICL performance on different NLP datasets using GPT-J-6B as our inference LLM. Here, we compare *Div-S3* against the baselines described in Sec. 5.3. Our proposed methods are marked with an asterisk (*). The subscripts represent the standard deviation corresponding to baselines repeated using three random seeds.