Benchmarking Procedural Language Understanding for Low-Resource Languages: A Case Study on Turkish

Arda Uzunoğlu[‡*], **Gözde Gül Şahin**[†]

[‡]Computer Science Department, Johns Hopkins University, Maryland, USA
[†]Computer Engineering Department, Koç University, Istanbul, Türkiye
[†]https://gglab-ku.github.io/

Abstract

Understanding procedural natural language (e.g., step-by-step instructions) is a crucial step to execution and planning. However, while there are ample corpora and downstream tasks available in English, the field lacks such resources for most languages. To address this gap, we conduct a case study on Turkish procedural texts. We first expand the number of tutorials in Turkish wikiHow from 2,000 to 52,000 using automated translation tools, where the translation quality and loyalty to the original meaning are validated by a team of experts on a random set. Then, we generate several downstream tasks on the corpus, such as linking actions, goal inference, and summarization. To tackle these tasks, we implement strong baseline models via fine-tuning large language-specific models such as TR-BART and BERTurk, as well as multilingual models such as mBART, mT5, and XLM. We find that language-specific models consistently outperform their multilingual models by a significant margin across most procedural language understanding (PLU) tasks. We release our corpus, downstream tasks and the baseline models with https://github.com/GGLAB-KU/turkish-plu.

1 Introduction

A procedural text typically comprises a sequence of steps that need to be followed in a specific order to accomplish a goal. For example, to care for an indoor plant, one must undertake tasks such as i) *selecting an appropriate location for the plant*, ii) *maintaining indoor humidity levels*, and iii) *selecting the right fertilizer*, usually in the given order. To accomplish a goal given with step-by-step instructions, a set of diverse skills that can be related to traditional NLP tasks such as semantic analysis (e.g., who did what to whom), commonsense reasoning (e.g., plant requires water), and coreference resolution (e.g., *it* refers to the *plant*) are required. Hence, procedural language understanding (PLU) can be considered a proxy to measure the performance of models on a combination of these distinct skills.

Previous work has immensely utilized the WikiHow tutorials, and proposed several downstream tasks on procedural text. For example, Zhang et al. (2020b) introduced step and goal inference tasks where the objective is to predict the most likely *step* given the *goal* or vice versa. Similarly, Zellers et al. (2019) proposed predicting the *next event* given the goal and the current step. All of these tasks are formulated as multiple-choice QA and require a partial understanding of step-goal relations in procedural documents. Furthermore, Zhou et al. (2022) proposed an information retrieval task where the goal is to link *steps* to related *goals* to create a wikiHow hierarchy. Finally, several other works (Koupaee and Wang, 2018; Ladhak et al., 2020) proposed an abstractive summarization task, that requires competitive language generation skills.

Despite its importance, PLU has been largely ignored for the majority of the languages due to a lack of language-specific web corpora. Except from Ladhak et al. (2020), all the aforementioned tasks are only available in English. In addition to the scarcity of raw text, creating downstream task data is challenging and might require language-specific filtering techniques to ensure high quality. Finally, all previous works study the proposed tasks in isolation, which can only give a limited insight into the model's performance.

Considering the uneven distribution of available procedural data across languages[1], our objective is to inspire research efforts on PLU for other understudied languages from different language families. To achieve this, we design a case study focused

[*]The work was done while the first author was at Eskişehir Bahçeşehir College.

[1]Although wikiHow comprises 19 languages, only two languages (English and Spanish) have more than 100k articles in parallel (Ladhak et al., 2020).

Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), pages 804–819
November 1–4, 2023.

on the Turkish language. Unlike previous works, we adopt a centralized approach and introduce a comprehensive benchmark that contains six downstream tasks on procedural documents.

To address the scarcity of resources, we utilize automatic machine translation tools. We implement rigorous quality control measures for machine translation including human evaluation, and show that the data is indeed high-quality. Next, we survey and study several downstream tasks and create high-quality, challenging task data through language-specific filtering and manual test data annotation. Finally, we perform a comprehensive set of experiments on a diverse set of language models with different pretraining, fine-tuning settings, and architectures. We find that language-specific models mostly outperform their multilingual counterparts; however, the model size is a more important factor than training language, i.e., large enough multilingual models outperform medium sized language-specific models. We show that tasks where we can perform rigorous language-specific preprocessing such as goal inference, are of higher-quality, thus more challenging. Finally, we find that our best-performing models for most downstream tasks, especially reranking, goal inference, and step ordering, are still far behind their English counterparts, suggesting a large room for improvement. We release all the resources—including the structured corpus of more than 52,000 tutorials, data splits for six downstream tasks and the experimented baseline models— at `https://github.com/GGLAB-KU/turkish-plu`.

2 Related Work

WikiHow is an eminent source for studying procedural text, allowing for a broad range of NLP tasks to be proposed and studied, such as linking actions (Lin et al., 2020; Zhou et al., 2022), step and goal inference (Zhang et al., 2020b; Yang et al., 2021), step ordering (Zhang et al., 2020b; Zhou et al., 2019), next event prediction (Nguyen et al., 2017; Zellers et al., 2019), and summarization (Koupaee and Wang, 2018; Ladhak et al., 2020). While these works serve as a proxy to procedural text understanding, they are mostly limited to English.

Exploiting machine translation tools is a common practice to generate semantic benchmarks for many resource-scarce languages. For instance, Mehdad et al. (2010) automatically translated hypotheses from English to French to generate a textual entailment dataset. Similarly, Real et al. (2018) created a Portuguese corpus for natural language inference (NLI), namely as SICK-BR, and Isbister and Sahlgren (2020) introduced the first Swedish benchmark for semantic similarity, by solely employing automatic translation systems. Moreover, Budur et al. (2020) and Beken Fikri et al. (2021) employed Amazon and Google translate to generate Turkish NLI and sentence similarity, datasets via automatically translating existing resources such as SNLI (Bowman et al., 2015), MNLI (Williams et al., 2018) and STS-B (Cer et al., 2017).

3 Turkish PLU Benchmark

To evaluate the procedural language understanding capacity of existing models and to improve upon them, we introduce i) a large procedural documents corpus covering a wide range of domains for Turkish, ii) a diverse set of downstream tasks derived from the corpus to evaluate distinct large language models and iii) strong baselines for each task.

3.1 Corpus

Following previous work (Zhang et al., 2020b), we utilize wikiHow, a large-scale source for procedural texts that contains how-to tutorials in a wide range of domains, curated by experts. We follow the format used by Zhang et al. (2020b) and extract the title, methods/parts, steps, and additional information, such as the related tutorials, references, tips, and warnings. We focus on the categories with the least subjective instructions (e.g., Crafts) and ignore subjective categories (e.g., Relationships).

Our corpus creation process has two steps: i) scraping the original Turkish wikiHow, and ii) translating the English tutorials from the English wikiHow corpus (Zhang et al., 2020b).

Scraping Turkish Wikihow Using the beautifulsoup library (Richardson, 2007), we scrape the Turkish wikiHow tutorials from the sitemap files. After the category filtering and deduplication process, we get over 2,000 tutorials.

Translating the English Wikihow To automatize the translation process, we first develop an open-source *file-level* translation tool: ÇEVERI. It is simply an easy-to-use Google Translate[2] wrapper that utilizes recursive search to find, translate

[2]https://cloud.google.com/translate

BLEU	ROUGE	METEOR	COMET	chrF	chrF++
23.51	52.25	44.32	88.12	67.91	62.08

Table 1: BLEU, ROUGE, METEOR, COMET, chrF, and chrF++ scores calculated over 1734 translated English-Turkish article pairs. All of the metrics are mapped to the interval of [0, 100] for convenience. Higher score indicates better translation for each evaluation metric.

	Fleiss' Kappa	Average	Agree 5	Agree +4
i)	0.751	4.40	47%	69%
ii)	0.813	4.76	78%	87%

Table 2: Results of the expert human validation on automatic machine translation quality control. Agree 5 and +4 respectively represent the percentage of the experts who agree that the score must be 5 or 4 and more.

and replace nested text fields within a file (see Appendix D). After filtering the subjective categories, we translate over 50,000 tutorials using ÇEVERI.

MT Quality Control To measure the translation quality of ÇEVERI, we translate the English counterparts of the original Turkish wikiHow tutorials and calculate a set of automatic evaluation metrics such as BLEU and COMET (Papineni et al., 2002; Lin, 2004; Banerjee and Lavie, 2005; Rei et al., 2020; Popović, 2015) given in Table 1. Although we use conventional metrics such as BLEU to align well with the literature, we are aware of the concerns related to them (Freitag et al., 2022). Therefore, we include metrics that better correlate with human evaluations, such as COMET (Mathur et al., 2020; Freitag et al., 2021), and consider character-level information such as chrF (Popović, 2015). Considering these, ÇEVERI achieving considerably high COMET and chrF scores indicate that the translation is, indeed, of high quality.

We also conduct human validation with three native Turkish speakers fluent in English. We randomly sample 104 step triplets: a) the original Turkish step, b) the corresponding English step, and c) the translation of the English step with respect to the category distribution of our corpus. Each expert is asked to evaluate the triplets by i) scoring the translation quality with the English step and the translated Turkish step and ii) scoring the semantic similarity between the original and the translated Turkish steps both between 1 and 5 (inclusive; 5 is the best). As given in Table 2, the results are highly reassuring, indicating high average scores with substantial agreement (Fleiss, 1971). Additionally, we perform a pilot study to investigate the feasibility of using machine-translated data and find that silver data bring a noticeable improvement (see Appendix E). Therefore, we consider the automatically generated part of our corpus to be of high quality due to the results of both the automatic and manual quality controls and the pilot study.

Source	#Tutorials	#Steps Avg Steps	#Methods Avg Methods
C&OV	2K	32K 13.42	5K 2.33
C&E	16K	229K 13.89	34K 2.10
HE	11K	154K 14.34	31K 2.87
H&C	9K	119K 13.37	19K 2.20
H&G	10K	133K 13.66	25K 2.59
P&A	4K	53K 13.75	11K 2.86
Original	2K	38K 19.15	7K 3.35
Translated	50K	681K 13.61	120K 2.40
Total	**52K**	**719K** **13.83**	**127K** **2.43**

Table 3: Final corpus statistics. C&OV: Cars and Other Vehicles, C&E: Computers and Electronics, HE: Health, H&C: Hobbies and Crafts, H&G: Home and Garden, P&A: Pets and Animals. Avg Step and Method: Average number of steps and methods per tutorial, respectively. A method is a set of steps that can be followed to achieve the given goal, while a step is a single instruction.

Corpus Statistics Our final corpus has more than 52,000 tutorials from six wikiHow categories, which contain around 719K steps and around 127K methods, with an average of 13.83 steps and 2.43 methods per tutorial as given in Table 3. Computers and Electronics is the largest category, while the Cars and Other Vehicles is the smallest. We posit the number of tutorials for a category decreases as the level of expertise needed for writing tutorials for that category increases. Health category is an exception to this, as most of its articles do not really go into depth, and contain basic and simple instructions. Although average numbers of steps and methods per tutorial are consistent by categories, they vary by data creation methods. We believe the reason for such a difference is that the tutorials translated and added to Turkish wikiHow by editors are far more popular and gripping tutorials, which probably correlates with the level of ease, thus the descriptiveness and comprehen-

Task	Train	Validation	Test
Linking Actions	1319	—	440
Goal Inference	255K	5K	837
Step Inference	124K	5K	612
Step Ordering	539K	10K	1021
Next Event Prediction	82K	5K	656
Summarization	113K	6K	6K

Table 4: Downstream tasks and dataset split sizes.

siveness, of the tutorials. We hypothesize that they are prioritized in the translation line by wikiHow editors, as they attract more attention.

3.2 Downstream Tasks

Next, we inspire from previous works that studied a single downstream task created on wikiHow and combine them under a single benchmark, summarized in Table 4 and explained below.

Linking Actions The task is defined as detecting the links between the steps and the goals across articles as shown in Figure 1. The steps provided in the tutorials, along with their hyperlinked goals, serve as the ground-truth data for the linking actions task.

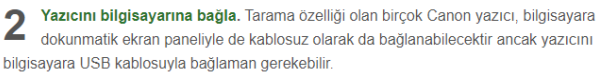

Figure 1: An example step with a hyperlink redirecting it to a tutorial. (Step says “Connect your printer to your computer" and the redirected tutorial has the title of “How to Connect a Printer to a Computer")

Goal Inference The goal inference task is simply defined as predicting the most likely goal, given a step. This task is structured as a multiple-choice format (Zhang et al., 2020b). For instance, when the prompt step is "Kıyafetlerini sık, böylece daha hızlı kuruyacaktır. (Squeeze your clothes, they would get dry quicker this way.)" and the candidate goals are:
A. Lavanta Nasıl Kurutulur? (How to Dry Lavender)
B. Kıyafetler Elde Nasıl Yıkanır? (How to Hand-Wash Clothes)
C. Kıyafetler Çabucak Nasıl Kurutulur? (How to Dry Clothes Quickly)
D. Islak Bir iPhone Nasıl Kurutulur? (How to Dry a Wet iPhone)
then the answer would be **C**.

We collect the positive step-goal pairs by iteratively picking them from each tutorial. For the negative candidate sampling, we consider both the semantic similarity with the positive candidate and the contextual plausibility for the step. We first encode each step in our corpus by averaging the BERT embeddings (Devlin et al., 2019) of the verb, noun, and proper noun tokens [3] contrary to Zhang et al. (2020b), which only considers the verb tokens. The reason why we include the additional POS tags is that most of the steps and goals in our corpus contain auxiliary verbs, which are common to Turkish such as *“yemek yapmak”* (to cook)[4]. Although contextualized embeddings help distinguish such differences to a certain extent, we observe that the incorporation of the additional parts brings a significant improvement in our negative candidate sampling strategy. Using FAISS (Johnson et al., 2021) with the our vector representations, we choose the top-3 goals with the highest cosine similarity to the positive candidate as the negative candidates. After the positive and negative candidate sampling, we randomly reassign one of the candidates as positive and correct the labels accordingly with a probability of 0.15 to avoid the model learning the sampling strategy. Lastly, we apply a set of hand-crafted filters (Zhang et al., 2020b) to ensure the quality of the task-specific data.

Step Inference Similar to the goal inference task, step inference is defined as predicting the most likely goal for the given step. It is also formulated as a multiple choice task (Zhang et al., 2020b). For instance, when the prompt goal is “Makas Nasıl Bileylenir? (How to Whet a Scissors)” and the candidate steps are:
A. Camı temizle. (Clean the glass/windows.)
B. Makası sil. (Wipe the scissors.)
C. Tuvaleti sil. (Wipe the toilet.)
D. Kartonu kes. (Cut the cardboard.)
the answer would be **B**.
We follow the same steps as in goal inference to sample positive and negative candidates by simply reversing the roles of the goals and the steps in the sampling process.

Step Ordering Here, the goal is to predict the preceding step out of the two given steps that help achieve a given goal. Similarly, it is formulated

[3] We conduct the POS tagging with the nlpturk library. https://github.com/nlpturk/nlpturk

[4] Such auxiliary verbs are mainly etmek, eylemek, olmak, kılmak and yapmak.

as a multiple-choice task. For instance, when the prompt goal is "YouTube'da Nasıl Yorum Bırakılır? (How to Leave a Comment on Youtube)" and the candidate steps are:
A. Bir video arayın. (Search for a Video.)
B. YouTube'u açın. (Open Youtube.)
B would be the answer since it must precede A.
For this task, we use the sampling strategy of (Zhang et al., 2020b). In wikiHow, some tutorials follow an ordered set of steps, while others contain alternative steps parallel to each other. Out of the ordered portion of our corpus, obtained in Appendix B, we use each goal as a prompt to sample step pairs with a window size of 1 and do not include any non-consecutive steps. We also randomly shuffle the pairs to prevent any index biases.

Next Event Prediction This task aims to produce the following action for a given context. It can be formulated as either a text generation task (Nguyen et al., 2017; Zhang et al., 2020a) or a multiple-choice task (Zellers et al., 2018, 2019). Following the formulation of the SWAG dataset (Zellers et al., 2018), we approach next event prediction task as a multiple-choice task, in which a model needs to predict the most likely continuation to a given setting out of the candidate events. For instance, when the prompt goal is "Sabit Disk Nasıl Çıkarılır? (How to Remove a Hard Drive)", the prompt step is "Bilgisayarın kasasını aç. (Open the Computer Case.)" and the candidate steps are:
A. Bilgisayar kasasının içinde sabit diski bul. (Locate the hard drive inside the computer.)
B. Bilgisayarının verilerini yedekle. (Back up your computer's data.)
C. Masaüstü anakartınla uyumlu bir sabit disk satın al. (Buy a hard drive that is compatible with your desktop motherboard.)
D. Windows yüklü bir masaüstü bilgisayarının olduğundan emin ol. (Make sure that you have a Windows desktop computer.)
then the answer would be **A**.
With the subgroup of our corpus labeled as ordered, we iteratively collect the prompt goals and two consecutive steps to use the prior step as the prompt step and the later step as the positive candidate. After obtaining the positive candidate, we use a similar sampling strategy that we used for goal inference. Unlike in goal inference, we additionally take pronoun token embeddings into account in order not to break the coreference chains.

Summarization Similar to Ladhak et al. (2020); Koupaee and Wang (2018), we formulate it as an abstractive summarization. We follow the data format proposed by Koupaee and Wang (2018) and build on the WikiLingua's (Ladhak et al., 2020) contributions to performing summarization over Turkish procedural text. Within the wikiHow platform, every step is composed of a concise headline resembling a summary and a descriptive paragraph providing detailed information about the step. In cases where tutorials lack methods or parts, we use the descriptions and headlines of the steps to form two distinct text bodies. These text bodies are then utilized to generate document-summary pairs. In the tutorials containing methods or parts, we follow a similar approach at the method or part level. An illustration of a step from the tutorial "Giysiden Küf Nasıl Çıkarılır? (How to Get Mold Out of Clothing)" is presented in Figure 2.

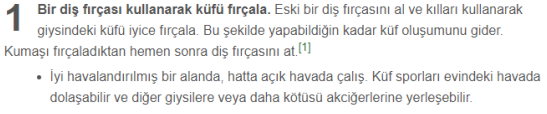
1 **Bir diş fırçası kullanarak küfü fırçala.** Eski bir diş fırçasını al ve kılları kullanarak giysindeki küfü iyice fırçala. Bu şekilde yapabildiğin kadar küf oluşumunu gider. Kumaşı fırçaladıktan hemen sonra diş fırçasını at.[1]
- İyi havalandırılmış bir alanda, hatta açık havada çalış. Küf sporları evindeki havada dolaşabilir ve diğer giysilere veya daha kötüsü akciğerlerine yerleşebilir.

Figure 2: An example step from the "How to Get Mold Out of Clothing" tutorial. The bolded part is the step headline, used as the summary, while the step description serves as the text to be summarized. The step description does not include the step headline, formulating the summarization task as the abstractive summarizaton.

3.3 Test Split Construction via Expert Annotation

Despite being synthetic, we incorporate examples from the machine-translated portion of our corpus into the test splits of our datasets. This decision stems from the limited availability of intersecting how-to tutorials on similar topics within the original Turkish wikiHow. Consequently, sampling negative candidates with high semantic similarity becomes challenging, leading to easily distinguishable positive candidates.

Due to the automated nature of our dataset creation process, some noise is present in the multiple choice task datasets. This noise includes false negative candidates and translations that are incorrect or ambiguous. For instance, consider the step "Yarayı tedavi etmeden önce ve sonra uygun el yıkama yapın. (Perform proper hand washing before and after treating the wound.)" which has a positive candidate of "Drenaj Yarasını Tedavi Etmek (Treat

a Draining Wound)" and a negative candidate of "Yatak Yaralarını Tedavi Etmek (Treat Bedsores)." While the negative candidate is sampled due to its high semantic similarity with the positive candidate, it is also a plausible option for the given step. To address this issue, we employ expert annotation to validate the test splits of the multiple choice datasets and eliminate such noisy examples.

We randomly sample 1000 examples for each of goal inference, step inference, and next event prediction tasks and 1500 examples for step ordering tasks, to be annotated by two experts. Firstly, the experts verify if there are multiple plausible candidates for each example. Secondly, the experts examine whether the translation has altered the meaning of any candidate. The annotation process results in approximately 60-80% of the randomly sampled examples, which are later utilized as the test splits, as illustrated in Table 4.

4 Models

Due to the distinct formulation of each task, we describe them individually below. For each task, we define the overall methodology. The implementation settings are described in Appendix G.

4.1 Linking Actions

We employ the retrieve-then-rerank strategy proposed by Zhou et al. (2022). As the name suggests, retrieve-then-rerank approach consists of two stages: i) Retrieval: the steps and goals are encoded in the dense embeddings space to perform semantic similarity search, and ii) Reranking: the top-n candidate goals are reranked for a given step by jointly encoding them.

During the retrieval stage, we initially encode the steps and goals individually. By obtaining embeddings of the steps and goals, we proceed to calculate the cosine similarity between pairs of goals and steps. Leveraging these computed cosine similarities, we employ semantic similarity search with FAISS (Johnson et al., 2021) to retrieve the top-n most similar candidates for each step. We experiment with both dense and sparse retrieval (e.g., BM25 (Robertson and Zaragoza, 2009)). For dense retrieval, we experiment with various sentence embedding models with different architectures (e.g., bi-encoder, cross-encoder), different fine-tuning data (e.g., NLI, STS, or both), and different pretraining data (e.g., Turkish or multilingual) described in details at Appendix A.1. In addition to existing sentence embeddings, we inspire by the recent success of the SimCSE architecture (Gao et al., 2021), and train our own Turkish-specific sentence embedding model, SimCSE-TR, in several training stages utilizing the text from Turkish Wikipedia and Turkish NLI (see Appendix C). Since each step has only one ground-truth goal, we use the standard recall metric to evaluate the retrieval models.

Encoding steps and goals independently is efficient; however, might result in information loss. Therefore, we rerank the top-n candidate list for each step, considering the step itself, the candidate goal, and the step's context, which includes surrounding steps or its goal. To accomplish this, we concatenate and input them into another model, utilizing the [CLS] token in the final hidden state to calculate a second similarity score. By reordering the top-n candidates based on the second similarity scores, we obtain the final list.

4.2 Multiple Choice Tasks

Since the goal inference, step inference, step ordering, and next event prediction tasks share a consistent formulation and adhere to the data format of the SWAG (Zellers et al., 2018) dataset, we employ an identical methodology across these tasks.

The models we investigate utilize a common strategy for the aforementioned multiple choice tasks. We provide the models with a question—the goal text for step inference and step ordering, the step text for goal inference, and both for next event prediction. Alongside the question, the models are given a candidate answer from the multiple options and generate a logit for that particular candidate. During the training process, we employ the cross-entropy loss to fine-tune our models, aiming to predict the correct candidate. We experiment with both Turkish-specific (i.e. BERTurk and DistilBERTurk (Schweter, 2020)) and multilingual (i.e. XLM (Conneau et al., 2020)) Transformer encoder models, as described in Appendix A.2. We use the standard metric, accuracy, to measure the performance. In addition to fine-tuning, we employ the models in a zero-shot setting.

4.3 Summarization

Safaya et al. (2022) introduces large pre-trained text generation models fine-tuned on the Turkish news summarization datasets, presenting out-of-domain baselines for summarization. We further fine-tune the aforementioned models to generate

the short descriptions (summaries) of the procedural tutorials (longer text bodies). We then test both the out-of-domain and in-domain procedural summarization models. Similarly, we experiment with both language-specific decoder models such as TR-BART (Safaya et al., 2022), and multilingual decoder models such as mBART (Liu et al., 2020) and mT5 (Xue et al., 2021), described in Appendix A.3. We use the standard ROUGE metrics for evaluation.

5 Results and Discussion

5.1 Linking Actions

We give the main results for both the retrieval and reranking models in Table 5. We observe that our SimCSE-TR models discussed in Appendix C outperform other baselines by a large margin. Furthermore, multilingual models generally perform worse than Turkish-specific models, which is expected. Similarly, XLM-R based models trained on parallel data for 50 languages (Conneau et al., 2020) generally perform worse than BERTurk-based models. Finally, we find that BM25 cannot be used in practical scenarios due to its low performance.

In the reranking stage, we introduce the ground-truth goal into the candidates' list, initially generated by the top-performing retrieval model. This addition occurs randomly after the 10th candidate, allowing us to assess the impact of reranking models. This modification significantly enhances the R@10 metric. However, it is noteworthy that DistilBERTurk exhibits a decline in R@1 performance, indicating that while it can distinguish the ground truth goals from other candidates, its improvement is limited to R@10. Conversely, BERTurk demonstrates a boost in both R@1 and R@10 performances.

The top-performing Turkish retrieval model achieves a comparable performance to the best-performing English retrieval model examined in Zhou et al. (2022). We attribute this similarity to the fact that the effectiveness of semantic similarity search remains consistent when the data and model quality levels are comparable across languages. However, it is worth noting that the best-performing Turkish reranking model exhibits a noticeable decline in performance compared to its English counterpart. We speculate that two factors contribute to this discrepancy: firstly, English dataset is significantly larger than Turkish dataset (21K vs. 1.7K), and secondly, the best-performing English reranking model, DeBERTa (He et al., 2021), is larger in size compared to the best-performing Turkish reranking model, BERTurk.

Model	R@1	R@10	R@30
XLM-R+NLI+STS	0.2	0.9	1.1
BM25	4.5	13.4	18.4
BERTurk+NLI+STS	9.3	17.3	24.3
Unsup. SimCSE-TR$_{\text{XLM-R}}$	11.6	24.5	33.9
XLM-R-XL-Paraphrase	15.9	33.0	41.1
S-XLM-R+NLI+STS	17.0	31.6	40.7
LaBSE	19.8	32.0	40.0
Sup. SimCSE-TR$_{\text{XLM-R}}$	25.9	42.7	54.1
S-BERTurk+NLI+STS	27.3	47.7	55.7
Unsup. SimCSE-TR$_{\text{BERTurk}}$	31.4	52.0	61.4
Sup. SimCSE-TR$_{\text{BERTurk}}$	**33.4**	**55.7**	**67.3**
+ DistilBERTurk	30.7	74.8	—
+ **BERTurk**	**40.5**	**78.9**	—

Table 5: The R@n indicates the percentage of the ground-truth goal being in the top-n candidates for a given step. The last two rows show the performances of the reranker models after including the gold goals in top-30 candidates generated by the best performing model, while the rest is retrieval only. We discuss the baseline models in Appendix A.

5.2 Multiple Choice Tasks

We observe a common pattern for the goal inference, step inference, and next event prediction tasks[5]: BERTurk performs the best, XLM-R is a close runner-up to the BERTurk, and DistilBERTurk performs slightly worse than XLM-R, as given in Table 6. In step ordering, DistilBERTurk performs slightly better than XLM-R.

Zero-shot performances of these models are on par with the random chance of guessing correctly, which means they cannot inherently understand the relationships between goal and step pairs, as well as step and step pairs. Furthermore, zero-shot performances of XLM-R are noticeably worse than those of BERTurk and DistilBERTurk. We believe this is due to the multilingual nature of XLM-R, which is not specialized in Turkish, unlike BERTurk and DistilBERTurk.

Significant improvements are observed with the fine-tuned models. The fine-tuned XLM-R model outperforms the fine-tuned DistilBERTurk model in all multiple choice tasks, except for step ordering. This observation suggests that the XLM-R model not only enhances its ability to select the correct

[5]While we manually check the performances of models with different random seeds, we only report the best run for all models, since the observed variances among different runs are small and would not cause any change in the rankings.

candidate but also improves its understanding of the Turkish language through fine-tuning.

When comparing the performance of language-specific models trained on Turkish data to those trained on English data, noticeable differences are observed. Turkish models exhibit significantly lower performances in goal inference and step ordering tasks. We attribute these variations to the dissimilarity in our sampling strategy, as explained in §3.2. Our sampling strategy considers a broader range of parts of speech compared to the approach used by Zhang et al. (2020b), resulting in candidates that are more similar at the embedding level and thereby increasing the difficulty. Additionally, while the performance decreases in goal inference, there is a slight improvement in step inference. This can be attributed to the fact that goals typically consist of less diverse parts of speech, mostly composed of a noun and a verb. As a result, the candidates sampled for goal inference tend to be more similar at the embedding level compared to step inference candidates, which often include additional parts of speech such as adjectives and adverbs.

Although we do not practice adversarial filtering to create our next event prediction dataset, we believe our sampling strategy also presents its own challenges. While the results shared in Zellers et al. (2018, 2019) are significantly lower than those of our models, the leaderboards for SWAG[6] and HellaSwag[7] datasets show that the challenge adversarial filtering brings can be overcome. Considering these, our results given in Table 6 are significantly lower than their English counterparts, suggesting a large room for improvement.

Additionally, we evaluate out-of-domain performances of some best-performing models to better understand their abilities in procedural tasks and find out their performances are generalizable to a certain extent, as discussed in Appendix F.

5.3 Summarization

The results are given in Table 7. As anticipated, in the summarization task, models that are fine-tuned on procedural summarization data outperform their out-of-domain fine-tuned counterparts. However, the performance improvement observed is relatively modest. We attribute this to the fact that the out-of-domain models still possess a robust capability acquired through their prior training on news summarization tasks.

Additionally, the multilingual out-of-domain models demonstrate superior performance compared to the single Turkish-specific model, TR-BART. However, in the procedural summarization task, TR-BART exhibits a higher performance boost and performs marginally better than procedural mT5. Both out-of-domain and procedural mBART models outperform other models. We attribute this to substantial size difference of mBART, which gives it an advantage over the other models.

When taking into account the model sizes and their multilingual capabilities, we conclude that both the specialization to Turkish and larger model sizes contribute to the overall performance improvement. However, our analysis reveals that a substantial difference in size can compensate for the multilingual aspect. This is evident in the comparison between out-of-domain and procedural TR-BART and mBART models, as presented in Table 7.

Task	Goal Inference	Step Inference	Step Ordering	Next Event Prediction
Random	25.00	25.00	50.00	25.00
XLM-R ZS (125M)	22.70	23.86	42.90	25.65
DistilBERTurk ZS (66M)	25.81	24.51	47.01	27.02
BERTurk ZS (110M)	**26.52**	**27.45**	**49.46**	**32.82**
DistilBERTurk FT (66M)	66.19	85.78	70.13	83.66
XLM-R FT (125M)	69.30	87.42	68.17	85.95
BERTurk FT (110M)	**72.40**	**91.34**	**72.09**	**88.55**

Table 6: Zero-Shot and Fine-Tuned performances of XLM-R, DistilBERTurk, and BERTurk models on multiple choice tasks, evaluated using accuracy. FT indicates that the model is fine-tuned on the task-specific data and ZS indicates zero-shot performance.

Model	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**
TR-BART OOD (120M)	16.28	4.21	12.35
mT5-base OOD (220M)	17.09	4.53	13.05
mBART OOD (680M)	18.30	5.12	13.82
TR-BART PRO (120M)	19.59	5.64	13.68
mT5-base PRO (220M)	19.30	5.33	14.42
mBART PRO (680M)	**22.62**	**6.43**	**15.69**

Table 7: Out-of-Domain Fine-Tuned, and Procedural Fine-Tuned performances of TR-BART, mBART, and mT5-base models in summarization task.

6 Conclusion

PLU tasks encompass various skills such as semantic analysis, commonsense reasoning, and coreference resolution. However, PLU has been primarily explored in English and the scarcity of language-specific resources limits its study in other

[6] https://leaderboard.allenai.org/swag/submissions/public
[7] https://rowanzellers.com/hellaswag/

languages. To address this gap, we present a case study in Turkish and introduce a centralized benchmark comprising six downstream tasks on procedural documents. We leverage machine translation tools and implement stringent quality control measures. We curate high-quality task data through language-specific filtering and manual annotation. Our experiments reveal that language-specific models tend to outperform multilingual models, but the model size is a critical factor. Tasks that involve rigorous language-specific preprocessing, such as goal inference, prove to be more challenging. Despite advancements, our best-performing models still lag behind their English counterparts, indicating large room for improvement. We release all resources publicly for further research.

Limitations

Our corpus creation method heavily relies on the success of the machine translation systems. However, such systems might have downfalls in specific cases. Local contexts and metrics are examples of such downfalls. We observe that some tutorials from the original Turkish wikiHow are localized, not directly translated. For instance, the Turkish counterpart of the tutorial titled "How to Lose 10 Pounds in 10 Days" is "10 Günde Nasıl 5 Kilo Verilir?" (How to Lose 5 Kilograms in 10 Days). In our case, Google Translate cannot distinguish these nuances.

Since the translated portion of our corpus makes up the majority, our models might pick up the translation artifacts, which, in turn, diminishes their success in actually learning their objective tasks.

mBART and mT5 models might generate biased summarizations, since they are previously trained on multilingual data and then fine-tuned on news summarizations before being fine-tuned on procedural documents.

The heavyweight fine-tuning and inference of mBART and mT5 sets a natural limitation to their usage. However, we overcome this limitation by practicing lightweight alternative solutions, such as half precision floating point format (FP16) training, optimization libraries, and gradient accumulation and checkpointing[8].

Lastly, the method we propose for the creation of procedural corpora in low-resource languages is implicitly dependent on the amount of resources for a language. This is because machine translation systems might not work in some low-resource languages as well as they work for Turkish.

[8]To the best of our knowledge, mT5 models currently cannot be trained with gradient checkpointing.

Ethics Statement

We use the content of wikiHow, which allows for the usage of its content under limited specific circumstances within the Creative Commons license. We abide all the conditions required by the Creative Commons license. The requirements of the Creative Commons also make the usage of English wikiHow corpus that we translate possible.

Since the source of the majority of our corpus and datasets are from translated tutorials, they might contain implicit biases due to the translation. Consequently, models trained on such data are also vulnerable to these biases.

Acknowledgements

This work has been supported by the Scientific and Technological Research Council of Türkiye (TÜBİTAK) as part of the project "Automatic Learning of Procedural Language from Natural Language Instructions for Intelligent Assistance" with the number 121C132. We also gratefully acknowledge KUIS AI Lab for providing computational support. We thank our anonymous reviewers and the members of GGLab who helped us improve this paper. We especially thank Shadi Sameh Hamdan for his contributions to setting up the implementation environment.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Figen Beken Fikri, Kemal Oflazer, and Berrin Yanikoglu. 2021. Semantic similarity based evaluation for abstractive news summarization. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 24–33, Online. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages

632–642, Lisbon, Portugal. Association for Computational Linguistics.

Emrah Budur, Rıza Özçelik, Tunga Gungor, and Christopher Potts. 2020. Data and Representation for Turkish Natural Language Inference. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8253–8267, Online. Association for Computational Linguistics.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard 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, Online. Association for Computational Linguistics.

Alexis Conneau and Douwe Kiela. 2018. SentEval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Joseph L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76:378–382.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, Eleftherios Avramidis, Tom Kocmi, George Foster, Alon Lavie, and André F. T. Martins. 2022. Results of WMT22 metrics shared task: Stop using BLEU – neural metrics are better and more robust. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 46–68, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondřej Bojar. 2021. Results of the WMT21 metrics shared task: Evaluating metrics with expert-based human evaluations on TED and news domain. In *Proceedings of the Sixth Conference on Machine Translation*, pages 733–774, Online. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Tim Isbister and Magnus Sahlgren. 2020. Why not simply translate? a first swedish evaluation benchmark for semantic similarity. *ArXiv*, abs/2009.03116.

J. Johnson, M. Douze, and H. Jegou. 2021. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(03):535–547.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset. *ArXiv*, abs/1810.09305.

Faisal Ladhak, Esin Durmus, Claire Cardie, and Kathleen McKeown. 2020. WikiLingua: A new benchmark dataset for cross-lingual abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4034–4048, Online. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021.

Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Angela Lin, Sudha Rao, Asli Celikyilmaz, Elnaz Nouri, Chris Brockett, Debadeepta Dey, and Bill Dolan. 2020. A recipe for creating multimodal aligned datasets for sequential tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4871–4884, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Nitika Mathur, Johnny Wei, Markus Freitag, Qingsong Ma, and Ondřej Bojar. 2020. Results of the WMT20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 688–725, Online. Association for Computational Linguistics.

Yashar Mehdad, Matteo Negri, and Marcello Federico. 2010. Towards cross-lingual textual entailment. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 321–324, Los Angeles, California. Association for Computational Linguistics.

Dai Quoc Nguyen, Dat Quoc Nguyen, Cuong Xuan Chu, Stefan Thater, and Manfred Pinkal. 2017. Sequence to sequence learning for event prediction. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 37–42, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Livy Real, Ana Rodrigues, Andressa Vieira e Silva, Beatriz Albiero, Bruna Thalenberg, Bruno Guide, Cindy Silva, Guilherme de Oliveira Lima, Igor C. S. Câmara, Miloš Stanojević, Rodrigo Souza, and Valeria de Paiva. 2018. Sick-br: A portuguese corpus for inference. In *Computational Processing of the Portuguese Language*, pages 303–312, Cham. Springer International Publishing.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.

Leonard Richardson. 2007. Beautiful soup documentation. *April*.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Ali Safaya, Emirhan Kurtuluş, Arda Goktogan, and Deniz Yuret. 2022. Mukayese: Turkish NLP strikes back. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 846–863, Dublin, Ireland. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Stefan Schweter. 2020. Berturk - bert models for turkish.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2020. MLSUM: The multilingual summarization corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8051–8067, Online. Association for Computational Linguistics.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4596–4604. PMLR.

Gugger Sylvain, Debut Lysandre, Wolf Thomas, Schmid Philipp, Mueller Zachary, and Mangrulkar Sourab. 2022. Accelerate: Training and inference at scale made simple, efficient and adaptable. https://github.com/huggingface/accelerate.

Andrew Trotman, Antti Puurula, and Blake Burgess. 2014. Improvements to bm25 and language models examined. In *Australasian Document Computing Symposium*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Wes McKinney. 2010. Data Structures for Statistical Computing in Python. In *Proceedings of the 9th Python in Science Conference*, pages 56 – 61.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Yue Yang, Artemis Panagopoulou, Qing Lyu, Li Zhang, Mark Yatskar, and Chris Callison-Burch. 2021. Visual goal-step inference using wikiHow. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2167–2179, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium. Association for Computational Linguistics.

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, Florence, Italy. Association for Computational Linguistics.

Hongming Zhang, Muhao Chen, Haoyu Wang, Yangqiu Song, and Dan Roth. 2020a. Analogous process structure induction for sub-event sequence prediction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1541–1550, Online. Association for Computational Linguistics.

Li Zhang, Qing Lyu, and Chris Callison-Burch. 2020b. Reasoning about goals, steps, and temporal ordering with WikiHow. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4630–4639, Online. Association for Computational Linguistics.

Shuyan Zhou, Li Zhang, Yue Yang, Qing Lyu, Pengcheng Yin, Chris Callison-Burch, and Graham Neubig. 2022. Show me more details: Discovering hierarchies of procedures from semi-structured web data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2998–3012, Dublin, Ireland. Association for Computational Linguistics.

Yilun Zhou, Julie Shah, and Steven Schockaert. 2019. Learning household task knowledge from WikiHow descriptions. In *Proceedings of the 5th Workshop on Semantic Deep Learning (SemDeep-5)*, pages 50–56, Macau, China. Association for Computational Linguistics.

A Baselines

A.1 Linking Actions

S-BERTurk + NLI + STS is the bi-encoder model that employs the Siamese and ternary network structures (Reimers and Gurevych, 2019) to derive close fixed-size sentence embeddings in vector space (Beken Fikri et al., 2021).

S-XLM-R + NLI + STS is the bi-encoder model that employs the Siamese and ternary network structures (Reimers and Gurevych, 2019) to derive close fixed-size sentence embeddings in vector space (Beken Fikri et al., 2021).

BERTurk + NLI + STS is the cross-encoder model that averages the BERT embeddings (Beken Fikri et al., 2021).

XLM-R + NLI + STS is the cross-encoder model that averages the XLM-R embeddings (Beken Fikri et al., 2021).

LaBSE stands for Language-agnostic BERT Sentence Embedding. It is trained on multilingual data for translation language modeling and produces sentence embeddings for 109 languages, including Turkish (Feng et al., 2022). We use the pretrained LaBSE model to generate Turkish sentence embeddings[9].

XLM-RoBERTA-base-XL-Paraphrase is a XLM-R model (Conneau et al., 2020) trained to imitate SBERT-paraphrases on parallel data for 50 languages (including Turkish) using multi-lingual knowledge distillation (Reimers and Gurevych, 2020). We use the pretrained XLM-RoBERTA-base-XL-Paraphrase model to generate Turkish sentence embeddings[10].

BM25 is a ranking function used to estimate the relevance between a set of documents to a given query based on the query terms appearing in each document (Robertson and Zaragoza, 2009). We use the BM25+ algorithm from the Rank-BM25 library[11], which implements the BM25 algorithms from (Trotman et al., 2014).

A.2 Multiple Choice Tasks

DistilBERTurk is the distilled version of its teacher model BERTurk, trained following the knowledge distillation method introduced by Sanh et al. (2019) (Schweter, 2020).

XLM-R is a transformer-based model trained on large multilingual data using the objective of multilingual masked language modeling (Conneau et al., 2020).

BERTurk is a transformer-based model trained on a combination of Turkish web corpora following the training methodology of Devlin et al. (2019) (Schweter, 2020).

[9]https://huggingface.co/sentence-transformers/LaBSE

[10]https://huggingface.co/sentence-transformers/paraphrase-xlm-r-multilingual-v1

[11]https://github.com/dorianbrown/rank_bm25

A.3 Summarization

TR-BART OOD is a Seq2Seq Transformer (Vaswani et al., 2017) trained on the Turkish split of the MLSUM dataset (Scialom et al., 2020) following the configuration of BART Base (Lewis et al., 2020)[12].

mBART OOD is a fine-tuned version of the pre-trained mBART50 (Liu et al., 2020). mBART50 is pre-trained on data from 50 different languages, and mBART OOD is fine-tuned on the Turkish split of MLSUM (Scialom et al., 2020)[13].

mT5-base OOD is a fine-tuned version of the pre-trained mT5-base (Xue et al., 2021). mT5-base is a multilingual variant of T5 (Raffel et al., 2020) that was pre-trained on a new Common Crawl-based dataset covering 101 languages, and mT5-base OOD is fine-tuned on the Turkish split of MLSUM (Scialom et al., 2020)[14].

B Classifying Orderliness of the Tutorials

wikiHow mostly contains two type of tutorials: i) tutorials with consecutive steps that must be followed in sequence (i.e. HDMI Televizyona Nasıl Bağlanır? (How to Connect HDMI to TV) has the steps Televizyonunda kullanılabilir bir HDMI girişi bul. (Locate an available HDMI port on your TV.), Doğru HDMI kablosunu al. (Get the right HDMI cable.), Kablonun bir ucunu cihaza bağla. (Connect one end of the cable to the device.)), ii) tutorials with steps that are parallel or alternative procedures to each other (i.e. Evde Ateş Nasıl Düşürülür? (How to Cure Fever at Home) tutorial has the steps Bol su iç. (Drink lots of water.), Rahat giysiler giy. (Wear comfy clothes.), and Oda sıcaklığını düşür. (Lower the room temperature.)).

Since the step ordering and next event prediction tasks require tutorials with ordered steps, we need to predict the orderliness of the tutorials in our corpus. First, expert authors annotate 900 tutorials based on the criteria of orderliness. With the obtained data, we fine-tune a BERTurk (Schweter, 2020) model for the binary text classification objective. Finally, we use it to classify each tutorial in our corpus, and use the tutorials labeled as ordered for the step ordering and next event predic-

[12]https://huggingface.co/mukayese/transformer-turkish-summarization

[13]https://huggingface.co/mukayese/mbart-large-turkish-summarization

[14]https://huggingface.co/mukayese/mt5-base-turkish-summarization

tion tasks. Our fine-tuned model's performance on our test split can be seen in Table 8.

Accuracy	Precision	Recall	F1
86.67	85.34	90.14	87.67

Table 8: Finetuned BERTurk's performance on our test split. We split the annotated 900 tutorials with the ratio of 70:15:15 (training:evaluation:test).

C SimCSE-TR

From using them to filter the goal and step inference tasks data to utilizing them in the retrieval stage of the linking actions task, we take advantage of sentence embeddings in a broad range. Therefore, we train a new Turkish-specific sentence embedding model utilizing the SimCSE architecture (Gao et al., 2021), which we name as SimCSE-TR.

SimCSE architecture employs a contrastive learning objective to derive meaningful sentence embeddings, with the hidden dropout mask acting as a minimal data augmentation method. In the unsupervised setting, SimCSE uses sentences from English Wikipedia to sample positive pairs by generating the representations of the same sentence with different dropout masks and negative pairs with the representations of different sentences. In the supervised setting, it integrates the annotated sentence pairs from natural language inference datasets into its contrastive training objective, utilizing the "entailment" pairs as positive pairs and "contradiction" pairs as hard negative pairs (Gao et al., 2021). Compared to other sentence embedding models and architectures, SimCSE converges faster with fewer data, which makes it lightweight to train and use. Furthermore, with a better aligned and more uniform latent space, it performs better on semantic textual similarity tasks and generates more distinguishable representation for sentences.

Following the implementation in SimCSE, we use randomly sampled one million sentences from Turkish Wikipedia for the unsupervised setting and the Turkish NLI datasets (Budur et al., 2020) for the supervised setting to train BERTurk (Schweter, 2020) and XLM-R based Turkish SimCSE models (Conneau et al., 2020). Similar to the English SimCSE, we train the unsupervised models for 1 epoch and the supervised models for 3 epochs. For each of the settings, we carry out a grid-search of batch size$\in\{64, 128, 256, 512\}$, learning rate$\in\{1e-5, 3e-5, 5e-5\}$, and maximum sequence length$\in\{16, 32, 64\}$ on Turkish STS-B development set, and report the best combinations in Table 9. We use the edited version of the SentEval (Conneau and Kiela, 2018) library shared in SimCSE Github repository[15] for the testing, and share the results in Table 10. Although they are not trained or fine-tuned on the train split of the Turkish STS-B, SimCSE-TR models perform comparably to other Turkish-specific sentence embedding models that are trained on Turkish STS-B.

Hyperparameter	Unsupervised		Supervised	
	BERTurk	XLM-R	BERTurk	XLM-R
Batch Size	64	512	512	512
Learning Rate	1e-5	1e-5	3e-5	5e-5
Max. Seq. Length	64	64	64	64

Table 9: Hyperparameters used in the training of SimCSE-TR models.

	Pearson	Spearman
Unsup SimCSE-TR XLM-R	66.23	66.95
Unsup. SimCSE-TR BERTurk	74.31	72.56
S-XLM-R♡ + NLI + STS	77.26	77.32
Sup SimCSE-TR XLM-R	79.70	80.30
Sup. SimCSE-TR BERTurk	79.07	81.06
XLM-R♡ + NLI + STS	81.94	81.21
S-BERTurk♡ + NLI + STS	82.85	83.31
BERTurk♡ + NLI + STS	**85.36**	**84.59**

Table 10: Performances of SimCSE-TR and other Turkish-specific sentence embedding models on the test split of the Turkish STS-B. ♡: taken directly from (Beken Fikri et al., 2021). Pearson and Spearman correlations were reported as $\rho \times 100$.

D ÇEVERI

ÇEVERI utilizes the pandas library (Wes McKinney, 2010) and recursive search to detect text values in seven different file format, .txt, .json, .xlsx, .csv, .xml, .pkl, and .docx. It, then, uses Google Translate to translate and replace detected texts. Although there is no usage-limit set by ÇEVERI, employment of the Google Translate makes it optimal to use ÇEVERI for a dataset consisting of a high number of smaller files, instead of a dataset consisting of a lower number of bigger files. Since it uses Google Translate in its backend, ÇEVERI can translate not only English but also all the languages Google Translate supports, as well as detecting and translating from unknown source languages.

[15] https://github.com/princeton-nlp/SimCSE

E Investigating the Feasibility of the Usage of Machine-Translated Data

In order to analyze the feasibility of using machine-translated data for studying procedural tasks, we conduct a pilot study in linking actions task.

First, we shuffle and recreate the train and test splits of our linking actions dataset. However, we do not include any silver data in the test split this time, contrary to what we did in §3.2. To test the practicability of using silver data, we incrementally increase the amount of the machine-translated data in the train split. We train the reranking models on these train splits with varying amounts of silver data and test them on the test split that solely consists of gold data. As seen in Figure 3, the utilization of silver data brings a noticeable improvement over the usage of only gold data to the performance of the reranking model. Furthermore, reranking models trained with a combination of gold and silver data outperforms the retrieval model consistently, on the contrary of reranking model trained with solely gold data underperforming the retrieval model in R@1 performance.

F Out-of-Domain Evaluation

To better understand the extent of our models' abilities in procedural tasks, we evaluate some of the best-performing models across other tasks.

Since the system needs to bring a continuation to the given set of actions in the next event prediction task, we hypothesize that next event prediction task implicitly covers the step inference task. In this regard, we believe that next event prediction models learn the relationship between the goals and steps, because the following actions to a given context must simultaneously serve the given goal. To investigate our claim, we test the BERTurk Next Event Prediction model on the test split of our step inference task. As Table 11 shows, BERTurk Next Event Prediction model achieves much higher performances than all the zero-shot models and the random probability.

To further examine the relationship between the next event prediction and step inference tasks, we also test the BERTurk Step Inference model on the test split of our next event prediction task. As Table 11 shows, BERTurk Step Inference model outperforms all zero-shot performances and the random probability, and performs closely to fine-tuned DistilBERTurk NEP, XLM-R NEP, and BERTurk NEP models.

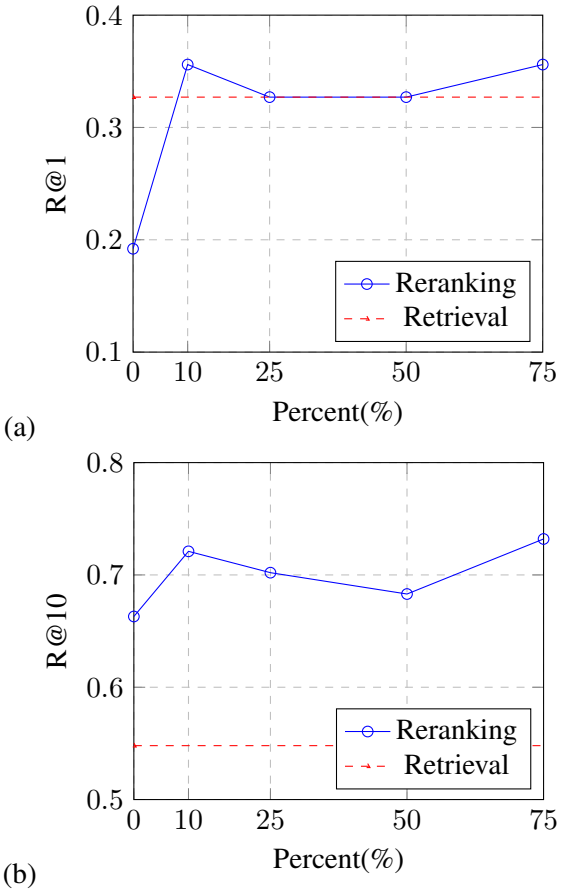

Figure 3: Performances of the BERTurk-based reranking models trained with different percentages of the translated data's train split. a) shows the performance change on R@1 and b) on R@10. 0% means reranking model is trained only with the originally Turkish data.

We believe the lower performance of BERTurk NEP on step inference data than the performance of BERTurk SI on next event prediction data is because BERTurk NEP is fine-tuned in a way that makes it dependent on the context information, which is absent in the step inference task. Similarly, we believe BERTurk SI obtains higher scores on next event prediction data than does BERTurk NEP on step inference data, because next event prediction task provides the context information, which might ease the objective of choosing the positive candidate.

G Implementation Details

We implement the reranking models as they are in Zhou et al. (2022)'s Github repository[16] with the same training setting. Since the dataset is small, training of the reranking models are quite lightweight, taking 15 to 45 minutes to train.

[16] https://github.com/shuyanzhou/wikihow_hierarchy

Model	SI	NEP
Random	25.00	25.00
BERTurk Zero-Shot	27.45	32.82
BERTurk NEP	61.93	88.55
BERTurk SI	91.34	80.46

Table 11: Performances of the best-performing Step Inference and Next Event Prediction models across step inference and next event prediction tasks. SI: Step Inference, NEP: Next Event Prediction.

We implement the summarization and multiple choice models using the Hugging Face libraries: Transformers (Wolf et al., 2020), accelerate (Sylvain et al., 2022), datasets (Lhoest et al., 2021), and evaluate. Transformers library enables us to work with the pre-trained models, accelerate library eases and accelerates the fine-tuning process and makes it more efficient, datasets library makes it easier to load and use datasets, and evaluate library facilitates the evaluation of the models.

With the accelerate library, we use FP16 training, gradient accumulation and checkpointing, and the Adafactor loss (Shazeer and Stern, 2018). This combination enables fine-tuning all the models on four NVIDIA T4s and test them on only one NVIDIA T4. In this setting, step inference and next event prediction models take 15 to 45 minutes, goal inference models take 30 to 90 minutes, step ordering models take 1 to 3 hours, and summarization models take approximately 9 to 18 hours to fine-tune.

Since we work with various models across different tasks, the hyperparameter setups for each dedicated task is given in details at https://github.com/GGLAB-KU/turkish-plu.