Learning to Select from Multiple Options

Jiangshu Du[1], **Wenpeng Yin**[2], **Congying Xia**[3], **Philip S. Yu**[1]

[1]University of Illinois at Chicago, Chicago, IL, USA
[2]Penn State University, State College, PA, USA
[3]Salesforce Research, Palo Alto, CA, USA
{jdu25, psyu}@uic.edu, wenpeng@psu.edu, c.xia@salesforce.com

Abstract

Many NLP tasks can be regarded as a selection problem from a set of options, e.g., classification tasks, multi-choice QA, etc. Textual entailment (TE) has been shown as the state-of-the-art (SOTA) approach to dealing with those selection problems. TE treats input texts as premises (P), options as hypotheses (H), then handles the selection problem by modeling (P, H) pairwise. Two limitations: (i) the pairwise modeling is unaware of other options, which is less intuitive since humans often determine the best options by comparing competing candidates; (ii) the inference process of pairwise TE is time-consuming, especially when the option space is large. To deal with the two issues, this work first proposes a contextualized TE model (Context-TE) by appending other k options as the context of the current (P, H) modeling. Context-TE is able to learn more reliable decision for the H since it considers various context. Second, we speed up Context-TE by coming up with Parallel-TE, which learns the decisions of multiple options simultaneously. Parallel-TE significantly improves the inference speed while keeping comparable performance with Context-TE. Our methods are evaluated on three tasks (ultra-fine entity typing, intent detection and multi-choice QA) that are typical selection problems with different sizes of options. Experiments show our models set new SOTA performance; particularly, Parallel-TE is faster than the pairwise TE by k times in inference.

Introduction

The NLP consists of various tasks and many of them are essentially a selection problem from a set of options. For instance, text classification selects the correct labels for the input text from all label candidates; given a paragraph and multiple answer candidates, multi-choice question answering (QA) selects the correct answer among all the choices. Textual entailment (TE) has been widely applied as the SOTA technique for solving these selection problems, e.g., intent detection (Xia et al. 2021), ultra-fine entity typing (Li, Yin, and Chen 2022), coreference resolution (Yin et al. 2020), relation extraction (Xia et al. 2021; Sainz et al. 2021), event argument extraction (Sainz et al. 2022), etc.

For those selection tasks, traditional classifiers treat all labels as indices, ignoring their semantics which, however, are the core supervision in zero- and few-shot learning. In contrast, TE reserves and exploits the original label information by constructing premise-hypothesis pairs, where premises (P) are texts and hypotheses (H) are transformed from labels. Then, TE selects an option by predicting if an option-oriented H can be entailed by the input P. In addition to using a unified textual entailment framework to solve various selection problems, another benefit is that the availability of large-scale entailment datasets, such as MNLI (Williams, Nangia, and Bowman 2018) and DocNLI (Yin, Radev, and Xiong 2021), can provide rich indirect supervision to handle those target problems when task-specific supervision is limited. However, there are two main limitations for the standard TE methods. First, TE pairs one input text only with a single option. Thus, all options are treated independently when the model makes decisions. In contrast, humans usually perform the selection in a more intuitive way: comparing all options and select the best one. Second, at the inference phase, the TE model needs to compare a text with all possible options one by one, which is inefficient especially when the option space is large. Take the ultra-fine entity typing (Choi et al. 2018) as an example, its 10k types take the TE model 35 seconds for each test instance and about 19.4 hours [1] to infer the entire test set (Li, Yin, and Chen 2022).

To overcome the limitations of standard TE, we introduce two novel approaches for option inference. First, inspired by the intuition that a model can be more powerful if it can find the correct answers even under the disturbance from other options, we propose a contextualized TE model (Context-TE). It appends other k options as an extra context of the standard (P, H) pairs during training. In this way, the model makes the decision not only depending on the current option H but also considering a more informative context. Thus, the prediction on H is more reliable, but Context-TE is as slow as the standard TE in terms of inference. Second, we improve the efficiency of Context-TE by introducing a parallel TE method (Parallel-TE). Parallel-TE learns an option's representation based on other options and makes the decisions of k options simultaneously. Therefore, the inference time of Parallel-TE is faster than Context-TE by k times.

We evaluate our proposed models on three tasks: ultra-fine entity typing (UFET (Choi et al. 2018), few-shot in-

[1]Experiments run at an NVIDIA TITAN RTX.

tent detection (BANKING77 (Casanueva et al. 2020), and multiple-choice QA (MCTest (Richardson, Burges, and Renshaw 2013)). The three tasks are representative selection problems: long texts and small-size options (MCTest), medium-size options (77 intents in BANKING77), and large-size options with multi-label selection (UFET has over 10,000 types and multiple can be correct for a given entity mention). Our proposed `Context-TE` sets the new state-of-the-art performance on all three tasks, and the `Parallel-TE` sacrifices a little bit performance (except for the UFET) while showing clear efficiency gains.

Our contributions can be summarized as the following three points. First, we discuss the limitations of the standard `TE` method in dealing with selection problems and propose two novel learning to select models—`Context-TE` and `Parallel-TE`—to overcome those issues. Second, our experiments show that both `Context-TE` and `Parallel-TE` outperform the standard `TE`, and set the new state-of-the-art performance on multiple benchmarks. Third, we provide a deep analysis to better understand why the new models work.

Related Work

Our work is mainly related with textual entailment and how it is applied to solve other NLP tasks.

Textual entailment. Dagan, Glickman, and Magnini (2006) first introduced the concept of `TE` and released a challenging benchmark, *Recognizing Textual Entailment* (RTE), for it. Then it attracted many following studies and the early stage of `TE` study mainly focused on lexical and syntactic level features (Androutsopoulos and Malakasiotis 2010; Rei and Briscoe 2011). In recent years, many large-scale `TE` datasets are released such as SNLI (Bowman et al. 2015), MNLI (Williams, Nangia, and Bowman 2018), SciTail (Khot, Sabharwal, and Clark 2018), ANLI (Nie et al. 2020) etc., which greatly advance the study of sentence-level `TE`. Yin, Radev, and Xiong (2021) also introduced a document-level dataset named DocNLI, which is constructed by reformatting and aggregating some NLP tasks. The recent `TE` systems mainly rely on pairwise modeling over the premise and hypothesis using attentive recurrent neural networks (Rocktäschel et al. 2016; Wang and Jiang 2016; Wang, Hamza, and Florian 2017), attentive convolutional neural networks (dos Santos et al. 2016; Yin and Schütze 2018), and pretrained transformers (Devlin et al. 2019; Liu et al. 2019). Our work is related to `TE` but more interested in applying `TE` to solve downstream selection problems.

Textual entailment solves other NLP tasks. Since many NLP tasks can be converted into a `TE` problem, `TE` naturally can provide indirect supervision for solving those tasks, especially when the task-specific supervision is limited. Yin, Hay, and Roth (2019) presented the first work that used `TE` supervision to solve zero-shot text classification. The idea was then applied to handle few-shot intent identification (Zhang et al. 2020; Xia et al. 2021), ultra-fine entity typing (Li, Yin, and Chen 2022), coreference resolution (Yin et al. 2020), relation extraction (Xia et al. 2021; Sainz et al. 2021), event argument extraction (Sainz et al. 2022), zero-shot machine comprehension (Yin, Radev, and Xiong 2021), etc. Wang et al. (2021) directly claimed that `TE` is a few-shot leaner for a wide range of NLP tasks.

All the literature we discussed above follow the standard `TE` method to solve selection problems, i.e., inferring the options one by one. This method suffers from two limitations as we stated in Introduction. In this work, we try to enhance the representation learning of `TE` and improve its inference speed in dealing with large option spaces.

Methods

This section presents our approaches to strengthening the options' representation learning (Section "`Context-TE`") and speeding up the selection process among large size of options (Section "`Parallel-TE`").

Problem Definition

We formulate the selection problem as follows: given a text t and an option space $\mathbb{O}$ consisting of n options $\{o_1, o_2, \cdots, o_n\}$, selecting the best (single-label) or more (multi-label) options that match the t under the definition of the task. Standard `TE` methods treat the t as the premise (`P`) and an option as a hypothesis (`H`). For each t, `TE` considers the option o_i without comparing o_i with other competing options. When the n is large, each t needs the pairwise modeling (t, o_i) totally n times in inference. In Section "`Context-TE`", we first present our method `Context-TE` to encode the interactions among options when modeling a particular option.

`Context-TE`

`Context-TE` is a contextualized representation learning paradigm for `TE`. It adds other options as the context when modeling the pair (`P`, `H`).

•**Contextualizing `TE` pairs.** `Context-TE` appends k options from $\mathbb{O}$ in a random order after the original (`P`, `H`) as a *competing context* ($c_1, c_2, \cdots, c_k$), resulting in a contextualized pair: (`P`, `H`, c_1, c_2, $\cdots$, c_k). Note that the options resulting in c_i ($i = 1, \cdots, k$) should not be the same as the option generating `H`. The competing context introduces the interactions between `H` and other k candidates in the option space in a single training instance so the model can better distinguish similar options. The gold entailment label (i.e., `entailment/contradict/neutral`) of the contextualized pair (`P`, `H`, $c_1, c_2, \cdots, c_k$) is set the same as (`P`, `H`). For instance, in an intent detection task, given an utterance text "*What's go on, where is my new card?*" and its gold intent "*card arrival*", a standard `TE` pair ("*What's go on, where is my new card?*", "*card arrival*") can be contextualized as ("*What's go on, where is my new card?*", "*card arrival*", "*lost or stolen card*", "*atm support*") if $k = 2$, and the relation between the option "card arrival" and the utterance "*What's go on, where is my new card?*" should not change despite the existence of competing context. The rationale of `Context-TE` is that if `P` can entail `H` in various contexts, `H` is more likely to be true given `P`.

•**Training.** As Figure 1 (middle) illustrates, we feed contextualized pairs into the encoder RoBERTa (Liu et al. 2019). The representation corresponding to the CLS token is used to

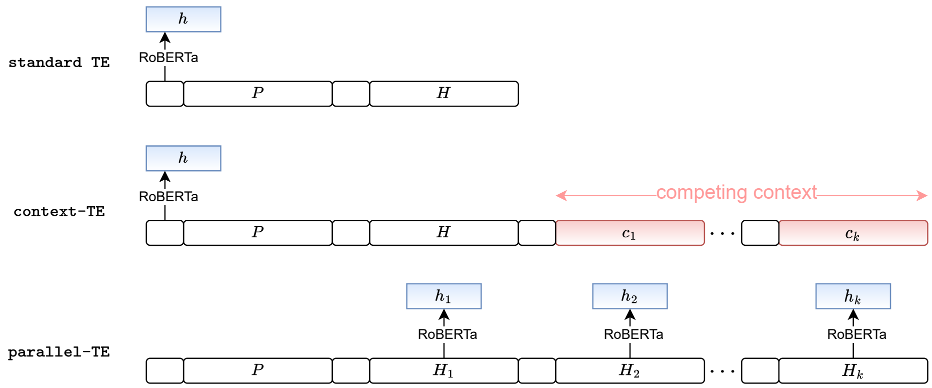

Figure 1: The comparison among TE, Context-TE and Parallel-TE in terms of their inputs and the representation (h) learning for hypotheses H.

denote the H in the context of P as well as the competing context ($c_1, c_2, \cdots, c_k$). The remaining classification architecture and training process are the same with standard TE that works on (P, H).

•**Inference.** The inference of Context-TE is the same as the standard TE. First, we convert the test set of the target task into (P, H) pairs by putting t with each option o_i together. Note that no competing context is needed for inference. Next, all the pairs are fed into the trained model and an entailment score for each pair will be given. For a single-label task, the option with the highest score will be returned. For a multi-label task, all the options with scores higher than a threshold τ are returned. Same as TE, Context-TE also needs to model $x \times n$ sizes of (P, H) pairs if there are x test inputs and n options. This time-consuming process motivates our second method Parallel-TE.

Parallel-TE

Parallel-TE also works on the contextualized pairs (P, H, c_1, c_2, $\cdots$, c_k) except that all options, including H and c_i ($i = 1, \cdots, k$), are treated equally and optimized to learn their respective labels simultaneously.

•**Training.** Different from Context-TE, all the options in the pair (P, H_1, $\cdots$, H_k) will keep their original labels (y_i for H_i) for joint training. Then the task is transformed to: given a sequence consisting of a premise and k hypotheses, select the correct hypotheses.

The input format of Parallel-TE is the same as Context-TE but they have different representation learning processes. As shown in Figure 1 (bottom), we first feed (P, H_1, $\cdots$, H_k) into RoBERTa and obtain a series of token-level representations on the top layer. Then for each H_i ($i = \{1, \cdots, k\}$), we average the representation vectors of all its tokens element-wise as its representation h_i:

$$h_i = \frac{1}{T}\sum_{j=1}^{T} \mathbf{RoBERTa}(\mathsf{H}_i^j), \tag{1}$$

where H_i has T tokens, and H_i^j is the j-th one.

Next, each H_i will learn a score s_i ranging from 0 to 1, indicating how likely H_i is entailed by P, by feeding h_i into a MLP:

$$s_i = \mathrm{sigmoid}(\mathrm{MLP}(h_i)) \tag{2}$$

The loss l of Parallel-TE is defined as the binary cross entropy (BCE) over all H_i ($i = 1, \cdots, k$):

$$l = -\frac{1}{k}\sum_{i=1}^{k} y_i \cdot \log s_i + (1 - y_i) \cdot \log(1 - s_i) \tag{3}$$

•**Inference.** For each hypothesis H_i in the input, Parallel-TE gives an entailment score. Then we can select the option of highest score for single-label tasks, or the options scored higher than a certain threshold for multi-label tasks. Because Parallel-TE models k hypotheses simultaneously for a single P, its inference is k times faster than the standard TE and Context-TE. More analyses are shown in Section "Inference speed of Parallel-TE".

Context-TE vs. Parallel-TE. They have the same input structures as illustrated in Figure 1. Both of them learn to select the correct option upon comparing with other options. They mainly differ in two aspects: i) Context-TE only models one hypothesis in each contextualized pair while Parallel-TE treats all the options in a pair as hypotheses, and infers all of them at the same time, which brings a huge boost regarding the inference efficiency; ii) Parallel-TE adopts BCE loss to deal with multi-label selection tasks. In contrast, Context-TE uses the cross entropy loss as each pair only needs to be classified as entailment or non-entailment.

For the real-world problems where many options exist such as ultra-fine entity typing and open-world intent detection, it is not feasible to embed all the options into a single pair due to the input length limit of the encoder. Therefore, we first retrieve top-k options to reduce the option space for those tasks. Details are discussed in Section " Experiments".

Experiments

Our experiments are conducted on three different tasks: *ultra-fine entity typing*, *few-shot intent detection* and *multiple-choice QA*. We choose the three tasks since they represent different selection problems in NLP. Ultra-fine entity typing is a multi-label task with a large option space: over 10,000 entity types. The few-shot intent detection task evaluates our proposed models under few-shot selection scene. Multiple-choice QA is a selection problem which requires the model to understand long paragraphs.

Top-k options generation. For the tasks with a large option space, it is infeasible to encode all options in the same input. We first find the top-k options for each text with an efficient method before the `Context-TE` and `Parallel-TE` steps, hoping that the top-k options are the most promising ones for the input text. The kept k options should have a high recall as we do not want the test inputs miss the gold options too much at this stage.

Concretely, for the ultra-fine entity typing task, we fine-tuned a bi-BERT (Devlin et al. 2019) on the training data with one BERT encoding an entity mention and another encoding a type candidate. For the intent detection task, we directly use the Sentence-BERT (Reimers and Gurevych 2019) to match an utterance and an intent candidate because utterances and their gold intents usually have high sentence similarities. The top-k model selection is based on the recalls on tasks' dev set. This step does not apply to the multi-choice QA task since there are only four answer candidates for each question.

The reason we choose bi-BERT or pretrained Sentence-BERT is that these representation learners decouple the `P` and `H` and use separate encoders to model them; as a result, they just need to generate representations for all options once and our model can reuse the same options' representations to compare with all inputs. After this step, each text t has k options that are most likely to be true. The top-k recall and the analysis about the influence of different k values are reported in Section "Influence of k values".

Model configurations. We use the public pretrained RoBERTa-large-mnli model as our backbone for the entity typing and intent detection tasks, and the pretrained RoBERTa-large on DocNLI (Yin, Radev, and Xiong 2021) for the QA task for a fair comparison. The hyperparameters, threshold τ, and k are searched on the dev set for each task.

Ultra-Fine Entity Typing

Dataset. In this task, we use the ultra-fine entity typing (UFET) benchmark (Choi et al. 2018). It has 5,994 human-annotated examples and 10,331 labels; each entity mention may have multiple types that are correct. The annotated examples are equally split into $train$, dev and $test$. In addition, UFET provides distant supervision data as an extra training resource, but we only train our models on the human-annotated dataset. The official evaluation metric is F1.

Baselines. The following prior systems are compared:

- **LDET** (Onoe and Durrett 2019) trains a learner to clean the distant supervision data. The learner discards the unusable data and fixes the noisy data. The denoised distant supervision data then is added to train a bi-LSTM (Hochreiter and Schmidhuber 1997) incorporated with pretrained ELMo (Peters et al. 2018) representations.
- **Box** (Onoe et al. 2021) leverages BERT to project entity mentions and types as box embeddings, which can better capture the hierarchical relationships between fine-grained entity types. Predictions are finalized according to the embedding intersection in the box embedding space.
- **LRN**. Liu et al. (2021) proposed a label reasoning network, leveraging auto-regressive networks and bipartite attribute graph to recognize the extrinsic and intrinsic dependencies among entity types. By exploiting the dependency knowledge and reasoning, more correct labels can be discovered at the inference stage.
- **MLMET** (Dai, Song, and Wang 2021) leverages the inner knowledge retained by a pretrained BERT model. The authors first used a BERT-based Masked Language Model (MLM) to generate extra distant supervision data and then trained a BERT model on three resources: the human-annotated, distant supervision, and MLM-generated data.
- **LITE** (Li, Yin, and Chen 2022) is the previous SOTA model. It converts the entity typing problem into a `TE` task and uses a RoBERTa-large model pretrained on MNLI (Williams, Nangia, and Bowman 2018) as the backbone. LITE treats entity-mentioning sentences as premises. Type options are first transformed to statements based on the template "[ENTITY] is a [LABEL]" and then are treated as hypotheses. It also adds label dependency in `TE`. A margin ranking loss is used to rank the entailment pairs over the non-entailment ones. *All `TE`-related models in this work, including `Context-TE` and `Parallel-TE`, use the same template as the baseline "LITE".*

Model	P	R	F1
LDET (Onoe and Durrett 2019)	51.5	33.0	40.1
Box (Onoe et al. 2021)	52.8	38.8	44.8
LRN (Liu et al. 2021)	**54.5**	38.9	45.4
MLMET (Dai, Song, and Wang 2021)	53.6	45.3	49.1
LITE (Li, Yin, and Chen 2022)	52.4	48.9	50.6
`Context-TE`	53.7	49.4	51.5
`Parallel-TE`	54.0	**51.0**	**52.4**

Table 1: Results on the UFET task.

Results. Table 1 shows the performance of our models and baselines. The k value is selected on the dev set and set to 80 for both `Context-TE` and `Parallel-TE`.

First, we notice that both our approaches `Context-TE` and `Parallel-TE` get new SOTA performance on this task with the `Parallel-TE` performs the best (52.4). This is particular impressive if we highlight that those baselines make use of either extra weak supervision data (e.g., LDET, MLMET) or the hierarchical relationships among types (e.g., Box, LRN, LITE). Second, even compared with LITE, the prior SOTA system that adopts the `TE` framework as well, `Context-TE` is able to learn better representations for hypotheses since it encoded the competing context.

Intent Detection

Dataset. We use the BANKING77 dataset (Casanueva et al. 2020) for the intent detection task. It is in the domain of online banking queries and consists of 77 intents. BANKING77 contains 10,003 training and 3,080 testing examples.

Model	5-shot	10-shot
CPFT (Zhang et al. 2021)	76.75	84.83
DualEnc. (Casanueva et al. 2020)	77.75	85.19
ConvB. (Mehri and Eric 2021)	-	85.95
DNNC (Zhang et al. 2020)	80.40	**86.71**
TE (Xia et al. 2021)	78.21	82.51
Context-TE	**80.76**	85.53
Parallel-TE	78.69	83.29

Table 2: Few-shot performance on BANKING77.

We explore the few-shot learning ability of our models in this task. From the training data, we randomly sample 5-shot and 10-shot instances per intent as our $train$ respectively. We also sample a small portion of the training dataset as our dev, following the previous setting (Zhang et al. 2021; Mehri, Eric, and Hakkani-Tür 2020). All experiments are run three times with distinctly sampled $train$ and we report the average performance, evaluated by accuracy.

Baselines. We compare with the following baselines:

- **DualEncoder** (Casanueva et al. 2020) is a dual sentence encoder model combined with USE (Cer et al. 2018) and ConveRT (Henderson et al. 2020). Both USE and ConverRT are strong sentence encoders pretrained on different conversational response tasks. The combination of both yields better performance than each of them.
- **ConvBERT+** (Mehri and Eric 2021) is based on ConvBERT (Mehri, Eric, and Hakkani-Tür 2020), a BERT model pretrained on a large dialogue corpus. By combining with example-driven training, task-adaptive training and observers, the model obtains a strong performance.
- **DNNC** (Zhang et al. 2020) is a discriminative nearest-neighbor model pretrained on three different TE datasets: SNLI (Bowman et al. 2015), MNLI (Williams, Nangia, and Bowman 2018), and WNLI (Levesque, Davis, and Morgenstern 2011).
- **CPFT** (Zhang et al. 2021) adopts RoBERTa as the backbone and first conducts self-supervised contrastive pretraining on six intent detection datasets. The model is then fine-tuned on few-shot data with supervised contrastive learning. *For a fair comparison, we report their performance without the contrastive pretraining on extra data.*
- **TE** (Xia et al. 2021) converts intent detection to traditional TE by treating utterances and intent options as premises and hypotheses, respectively.

Model details. Since the intent detection task is under the few-shot setting, data augmentation can be helpful during the training stage. Context-TE is naturally suitable for augmenting data because it expands the training data by introducing the extra (P, H) pairs equipped with competing context. For Parallel-TE, we perform data augmentation as follows: for each pair with k hypotheses, we shuffle it k times, making sure the positive option is at the different positions in all new sequences. This is to let the model learn an option's gold label wherever the option is located in the input sequence.

In this task, Context-TE infers at the full label space $\mathbb{O}$ ($|\mathbb{O}| = 77$). Due to the max sequence length limitation of RoBERTa, Parallel-TE infers only the top-k label options where k ranges from 10 to 60, selected on dev.

Results. Test accuracy under 5-shot and 10-shot settings on BANKING77 benchmark is reported in Table 2. Context-TE outperforms all the models under the 5-shot setting and obtains competitive performance with the prior SOTA on 10-shot (85.53 vs. 86.71). Parallel-TE demonstrates tiny drop of performance on both 5-shot and 10-shot. But compared with the standard TE, both Parallel-TE and Context-TE yield better results, suggesting that our proposed models are more effective than the traditional TE.

Multi-Choice Question Answering

Dataset. We work on MCTest (Richardson, Burges, and Renshaw 2013), a multiple-choice machine comprehension benchmark from a fictional story domain. One correct answer must be selected among four candidates given a question and a paragraph stating the background knowledge. Besides, Richardson, Burges, and Renshaw (2013) released a TE version data by treating paragraphs as premises and converting question-answer pairs into hypotheses. MCTest is a low-resource task, consisting of two sets with 160 (MC160) and 500 (MC500) examples, respectively. Our experiments are conducted on the TE version of MCTest. The official evaluation metric is accuracy.

Baselines. We compare with three latest methods:

- **RDEC** (Yu, Zha, and Yin 2019) designs an inferential network trained with reinforcement learning to understand contexts. This method consists of multiple attention-based reasoning steps and recursively constructs the evidence chain to improve the reasoning skill of machines.
- **TE**$_{\text{DocNLI}}$ (Yin, Radev, and Xiong 2021) is the previous SOTA system that first pretrains RoBERTa-large on DocNLI, a large corpus for document-level TE, then finetunes on MCTest.
- **TE**$_{\text{vanilla}}$ directly trains a RoBERTa-large on the TE version data without pretraining on any extra TE datasets such as MNLI, DocNLI.

Model details. In this task, Parallel-TE also performs data augmentation with the strategy elaborated in Section "Intent detection" because the training data is scarce. In addition, Context-TE follows TE$_{\text{DocNLI}}$ to exploit the pretrained RoBERTa on DocNLI as the encoder. Note that this task does not require the top-k option generation since the option size for each piece of data is merely four, but we allow the system to truncate the end of premises if the input length exceeds the max length limit of RoBERTa.

Results. Table 3 shows the experiment results on MCTest. Both Context-TE and TE$_{\text{DocNLI}}$ rely on the indirect supervision from DocNLI, but Context-TE outperforms TE$_{\text{DocNLI}}$ on both MC160 and MC500, achieving the new SOTA performance. Parallel-TE performs worse than Context-TE because it is more comparable with the TE$_{\text{vanilla}}$: both use the RoBERTa-large encoder and do not pretrain on extra data.

Model	MC160	MC500
TE$_{\text{vanilla}}$	42.50	63.67
RDEC (Yu, Zha, and Yin 2019)	80.00	75.50
TE$_{\text{DocNLI}}$ (Yin et al. 2021)	90.83	90.66
`Context-TE`	**92.50**	**91.67**
`Parallel-TE`	81.25	82.50

Table 3: Test accuracy on MCTest.

Model	UFET			BANKING77			
	P	R	F1	1-shot	3-shot	5-shot	10-shot
TE	-	-	-	67.53	73.74	78.21	82.51
TE$_{\text{top}-k}$	**54.4**	47.7	50.8	63.95	72.54	77.31	80.54
LITE$_{\text{top}-k}$	49.6	51.0	50.3	-	-	-	-
`Context-TE`	53.7	49.4	51.5	**69.40**	**77.42**	**80.76**	**85.53**
`Parallel-TE`	54.0	**51.0**	**52.4**	69.11	75.12	78.69	83.29

Table 4: Ablation study on both ultra-fine entity typing and intent detection tasks. [MODEL]$_{\text{top}-k}$ refers to the model runs on the retrieved top-k options. `TE` runs on the full option space.

Nevertheless, `Parallel-TE` achieves multi-option joint training, which leads to the improvement by large margins: 81.25 vs. 42.50 on MC160, and 82.50 vs. 63.67 on MC500.

Analysis

What Contributes to the Improvement?

The experimental results on the three benchmarks demonstrate the effectiveness of our proposed methods. However, it is still unclear: *does the system benefit from the top-k option generation as it reduces the option space or other inference strategies?* We conduct ablation study on both ultra-fine entity typing and intent detection tasks since our models need to perform the top-k option generation for them.

Specifically, given the top-k selected options, we run other competitive systems to see if the smaller option space helps them. For the entity typing task, we run the prior SOTA model, LITE, on the same top-k ($k = 80$) as our `Parallel-TE` model. For the intent detection task, we run the standard `TE` method on the same top-k ($k = 25$). Note that the `Context-TE` model infers in the entire option space on BANKING77. To gain a better insight into the impact of the top-k generation step, we also extend our experiments with 1-shot and 3-shot settings. The data sampling and evaluation strategies do not change.

The ablation experiments results are shown in Table 4. Given top-k options, both our systems `Context-TE` and `Parallel-TE` turn to be the TE$_{\text{top}-k}$ if we discard the competing context and the multi-option joint training. However, we notice that TE$_{\text{top}-k}$ gets pretty close performance with the LITE$_{\text{top}-k}$ on UFET (50.8 vs. 50.3 by F1) and even slightly worse performance than the standard `TE` on all the four few-shot settings of BANKING77. In contract, given the same top-k options, both `Context-TE` and `Parallel-TE` surpass the competitors with large margins. This means that our systems mainly benefit from the newly designed representation

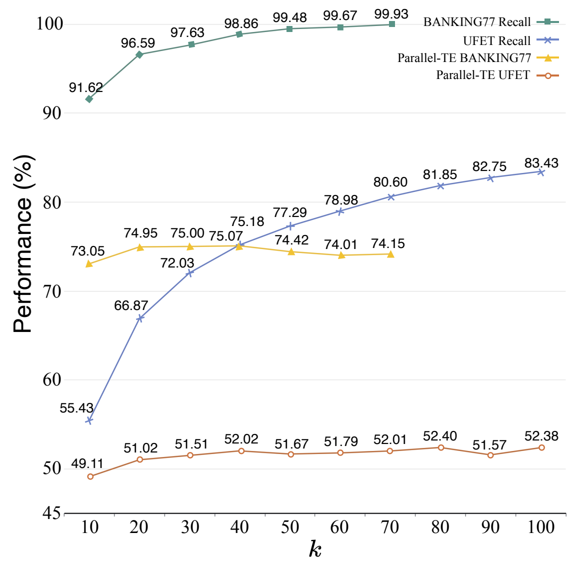

Figure 2: Top-k recall and `Parallel-TE` performance on UFET and BANKING77 tasks when k value varies. For the `Parallel-TE` performance, we report F1 and accuracy on UFET and BANKING77, respectively.

learning and training strategy rather than the smaller option space by top-k generation.

Influence of k Values

A higher k value can bring a higher top-k recall; however, it also caps the model performance. In this section, we investigate how the k values affects the `Parallel-TE` performance.

As shown in Figure 2, we report top-k recall, where k varies from 10 to 100 for UFET and 10 to 70 for (3-shot) BANKING77 due to the max length limit of RoBERTa. We also show the `Parallel-TE` performance under different k values. The top-k recall on BANKING77 is already high (91.62%) even when $k = 10$, and its increasing speed slows down when $k > 30$. `Parallel-TE` achieves the best performance with $k = 25$ as reported in the previous section. After that, a higher k harms the `Parallel-TE` performance. UFET top-k recall gets more benefits from the increase of k. The highest F1 is achieved when $k = 80$, and a very close performance is obtained when $k = 100$. These observations demonstrate a trade-off between k values and the model performance. As k increases, more correct options are captured in the top-k option space, but it also brings more difficulties for `Parallel-TE` to select the correct one since more competing options exist. Nonetheless, it is still beneficial to provide opportunities for those competing options to interact with each other so that the model can make better decisions.

Influence of Top-k Orders

We study this factor in both training and inference.

Training. As per our experiments, the order of top-k options in a input instance is essential to `Parallel-TE` at the training phase. We first start our experiments on UFET and

Model	1-shot	3-shot	5-shot	10-shot
Parallel-TE	69.11	75.12	78.69	83.29
w/ majority vote	**69.89**	**75.70**	**79.31**	**83.57**

Table 5: The results of Parallel-TE and the majority voting on BANKING77.

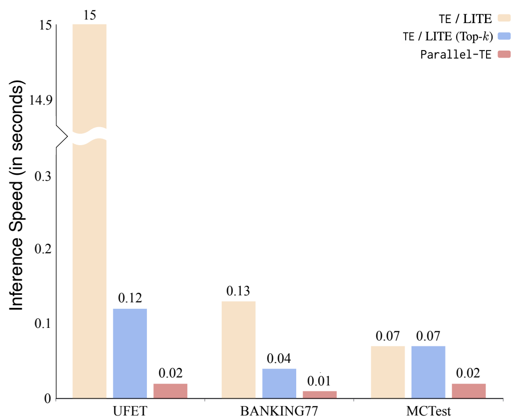

Figure 3: Inference speed (seconds/test case) of different TE models. TE / LITE (Top-k) indicates that the model runs on the top-k option space.

keep the original order of the top-k options. After top-k generation, most correct options are at the front part since they usually receive higher scores. Training a model on the pairs with the unshuffled top-k options leads to a serious overfitting on the position information. Thus, shuffling the order of top-k is important for Parallel-TE during training.

Inference. We also investigate the influence of different top-k orders at the inference stage by conducting experiments on few-shot BANKING77. Given the same text, different top-k orders sometimes yield predictions of tiny differences but mostly the predictions are consistent. This indicates that our training strategy in Parallel-TE can result in very robust model behavior. Duplicating a pair with the shuffled top-k orders and then making the final decision by majority voting improves the performance slightly, as shown in Table 5, but these negligible improvements are not worth the time cost.

Inference Speed of Parallel-TE

We compare the inference speeds of Parallel-TE with other entailment-based methods (TE, LITE and their respective versions working on top-k options) in Figure 3. [2]

Overall, the inference time of TE/LITE over the three tasks decreases in the order of "UFET $\rightarrow$ BANKING77 $\rightarrow$ MCTest" since both methods search the gold option from the whole space and each subsequent task has a smaller size of options. The top-k options can speed up TE/LITE (i.e., TE/LITE (top-k)) dramatically, which is within expectation. However, as we discussed in Section "What contributes to the improvement?", this operation may degrade the model performance.

[2]The inference speed is measured on an NVIDIA GeForce RTX 3090 with the evaluation batch size of 256.

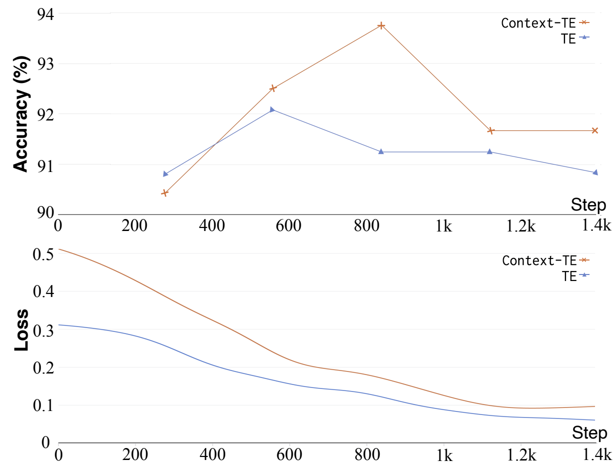

Figure 4: Training loss curves and test accuracy of Context-TE and TE on MC160 task. Loss curves are processed with Gaussian smoothing.

When apply our model Parallel-TE on the top-k options, the inference speed gets further boosted: on the UFET task, rerunning the prior SOTA model LITE takes 15 seconds to predict a single test instance, while our model Parallel-TE only spends 0.02 seconds.

Why Context-TE Works?

By equipping (P, H) pairs with competing context, Context-TE outperforms the standard TE model on all the three tasks. We try to explain it by analyzing the training loss curves and test accuracy of both models on MC160 task, as shown in Figure 4. We train both models 5 epochs and report the test accuracy at the end of each epoch. The training loss is recorded at every training step and processed with Gaussian smoothing. As the figure illustrates, Context-TE always outperforms TE after the second epoch while holding a higher training loss. This is reasonable because Context-TE introduces more challenging data (selecting the correct option under the disturbance of other options is harder), which acts as a regularizer that mitigates the overfitting to some extent.

Conclusion

This work studied the issues of the popular TE framework in solving selection tasks (i.e., *neglecting option-to-option comparison in representation learning* and *low inference speed*) and proposed Context-TE and Parallel-TE to solve them, respectively. Both new models outperform the standard TE method and mostly set the new state of the art performance in three typical tasks: ultra-fine entity typing, intent detection and multi-choice machine comprehension.

Acknowledgments

The authors appreciate the reviewers for their insightful comments and suggestions. This work is supported in part by NSF under grants III-1763325, III-1909323, III-2106758, and SaTC-1930941.

References

Androutsopoulos, I.; and Malakasiotis, P. 2010. A Survey of Paraphrasing and Textual Entailment Methods. *J. Artif. Intell. Res.*, 38: 135–187.

Bowman, S. R.; Angeli, G.; Potts, C.; and Manning, C. D. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of EMNLP*, 632–642.

Casanueva, I.; Temčinas, T.; Gerz, D.; Henderson, M.; and Vulić, I. 2020. Efficient Intent Detection with Dual Sentence Encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, 38–45.

Cer, D.; Yang, Y.; Kong, S.; Hua, N.; Limtiaco, N.; John, R. S.; Constant, N.; Guajardo-Cespedes, M.; Yuan, S.; Tar, C.; Sung, Y.; Strope, B.; and Kurzweil, R. 2018. Universal Sentence Encoder. *CoRR*, abs/1803.11175.

Choi, E.; Levy, O.; Choi, Y.; and Zettlemoyer, L. 2018. Ultra-Fine Entity Typing. In *Proceedings of ACL*, 87–96.

Dagan, I.; Glickman, O.; and Magnini, B. 2006. The PASCAL Recognising Textual Entailment Challenge. In Quiñonero-Candela, J.; Dagan, I.; Magnini, B.; and d'Alché Buc, F., eds., *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, 177–190.

Dai, H.; Song, Y.; and Wang, H. 2021. Ultra-Fine Entity Typing with Weak Supervision from a Masked Language Model. In *Proceedings of ACL-IJCNLP*, 1790–1799.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*, 4171–4186.

dos Santos, C. N.; Tan, M.; Xiang, B.; and Zhou, B. 2016. Attentive Pooling Networks. *CoRR*, abs/1602.03609.

Henderson, M.; Casanueva, I.; Mrkšić, N.; Su, P.-H.; Wen, T.-H.; and Vulić, I. 2020. ConveRT: Efficient and Accurate Conversational Representations from Transformers. In *Proceedings of EMNLP Findings*, 2161–2174.

Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Comput.*, 9(8): 1735–1780.

Khot, T.; Sabharwal, A.; and Clark, P. 2018. SciTaiL: A Textual Entailment Dataset from Science Question Answering. In *Proceedings of AAAI*, 5189–5197.

Levesque, H. J.; Davis, E.; and Morgenstern, L. 2011. The Winograd schema challenge. *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*, 46: 47.

Li, B.; Yin, W.; and Chen, M. 2022. Ultra-fine Entity Typing with Indirect Supervision from Natural Language Inference. *Transactions of ACL*, 10: 607–622.

Liu, Q.; Lin, H.; Xiao, X.; Han, X.; Sun, L.; and Wu, H. 2021. Fine-grained Entity Typing via Label Reasoning. In *Proceedings of EMNLP*, 4611–4622.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692.

Mehri, S.; and Eric, M. 2021. Example-Driven Intent Prediction with Observers. In *Proceedings of NAACL-HLT*, 2979–2992.

Mehri, S.; Eric, M.; and Hakkani-Tür, D. 2020. DialoGLUE: A Natural Language Understanding Benchmark for Task-Oriented Dialogue. *CoRR*, abs/2009.13570.

Nie, Y.; Williams, A.; Dinan, E.; Bansal, M.; Weston, J.; and Kiela, D. 2020. Adversarial NLI: A New Benchmark for Natural Language Understanding. In *Proceedings of ACL*, 4885–4901.

Onoe, Y.; Boratko, M.; McCallum, A.; and Durrett, G. 2021. Modeling Fine-Grained Entity Types with Box Embeddings. In *Proceedings of ACL-IJCNLP*, 2051–2064.

Onoe, Y.; and Durrett, G. 2019. Learning to Denoise Distantly-Labeled Data for Entity Typing. In *Proceedings of NAACL-HLT*, 2407–2417.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep Contextualized Word Representations. In *Proceedings of NAACL-HLT*, 2227–2237.

Rei, M.; and Briscoe, T. 2011. Unsupervised Entailment Detection between Dependency Graph Fragments. In *Proceedings of BioNLP@ACL*, 10–18.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings EMNLP-IJCNLP*, 3982–3992.

Richardson, M.; Burges, C. J.; and Renshaw, E. 2013. MCTest: A Challenge Dataset for the Open-Domain Machine Comprehension of Text. In *Proceedings of EMNLP*, 193–203.

Rocktäschel, T.; Grefenstette, E.; Hermann, K. M.; Kociský, T.; and Blunsom, P. 2016. Reasoning about Entailment with Neural Attention. In *Proceedings of ICLR*.

Sainz, O.; de Lacalle, O. L.; Labaka, G.; Barrena, A.; and Agirre, E. 2021. Label Verbalization and Entailment for Effective Zero and Few-Shot Relation Extraction. In *Proceedings of EMNLP*, 1199–1212.

Sainz, O.; Gonzalez-Dios, I.; de Lacalle, O. L.; Min, B.; and Agirre, E. 2022. Textual Entailment for Event Argument Extraction: Zero- and Few-Shot with Multi-Source Learning. In *Proceedings of NAACL-HLT*.

Wang, S.; Fang, H.; Khabsa, M.; Mao, H.; and Ma, H. 2021. Entailment as Few-Shot Learner. *CoRR*, abs/2104.14690.

Wang, S.; and Jiang, J. 2016. Learning Natural Language Inference with LSTM. In *Proceedings of NAACL-HLT*, 1442–1451.

Wang, Z.; Hamza, W.; and Florian, R. 2017. Bilateral Multi-Perspective Matching for Natural Language Sentences. In *Proceedings of IJCAI*, 4144–4150.

Williams, A.; Nangia, N.; and Bowman, S. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of NAACL-HLT*, 1112–1122.

Xia, C.; Yin, W.; Feng, Y.; and Yu, P. 2021. Incremental Few-shot Text Classification with Multi-round New Classes:

Formulation, Dataset and System. In *Proceedings of NAACL-HLT*, 1351–1360.

Yin, W.; Hay, J.; and Roth, D. 2019. Benchmarking Zero-shot Text Classification: Datasets, Evaluation and Entailment Approach. In *Proceedings of EMNLP-IJCNLP*, 3912–3921.

Yin, W.; Radev, D.; and Xiong, C. 2021. DocNLI: A Large-scale Dataset for Document-level Natural Language Inference. In *Proceedings of ACL-IJCNLP Findings*, 4913–4922.

Yin, W.; Rajani, N. F.; Radev, D.; Socher, R.; and Xiong, C. 2020. Universal Natural Language Processing with Limited Annotations: Try Few-shot Textual Entailment as a Start. In *Proceedings of EMNLP*, 8229–8239.

Yin, W.; and Schütze, H. 2018. Attentive Convolution: Equipping CNNs with RNN-style Attention Mechanisms. *Transactions of ACL*, 6: 687–702.

Yu, J.; Zha, Z.; and Yin, J. 2019. Inferential Machine Comprehension: Answering Questions by Recursively Deducing the Evidence Chain from Text. In *Proceedings of ACL*, 2241–2251.

Zhang, J.; Bui, T.; Yoon, S.; Chen, X.; Liu, Z.; Xia, C.; Tran, Q. H.; Chang, W.; and Yu, P. 2021. Few-Shot Intent Detection via Contrastive Pre-Training and Fine-Tuning. In *Proceedings of EMNLP*, 1906–1912.

Zhang, J.; Hashimoto, K.; Liu, W.; Wu, C.-S.; Wan, Y.; Yu, P.; Socher, R.; and Xiong, C. 2020. Discriminative Nearest Neighbor Few-Shot Intent Detection by Transferring Natural Language Inference. In *Proceedings of EMNLP*, 5064–5082.