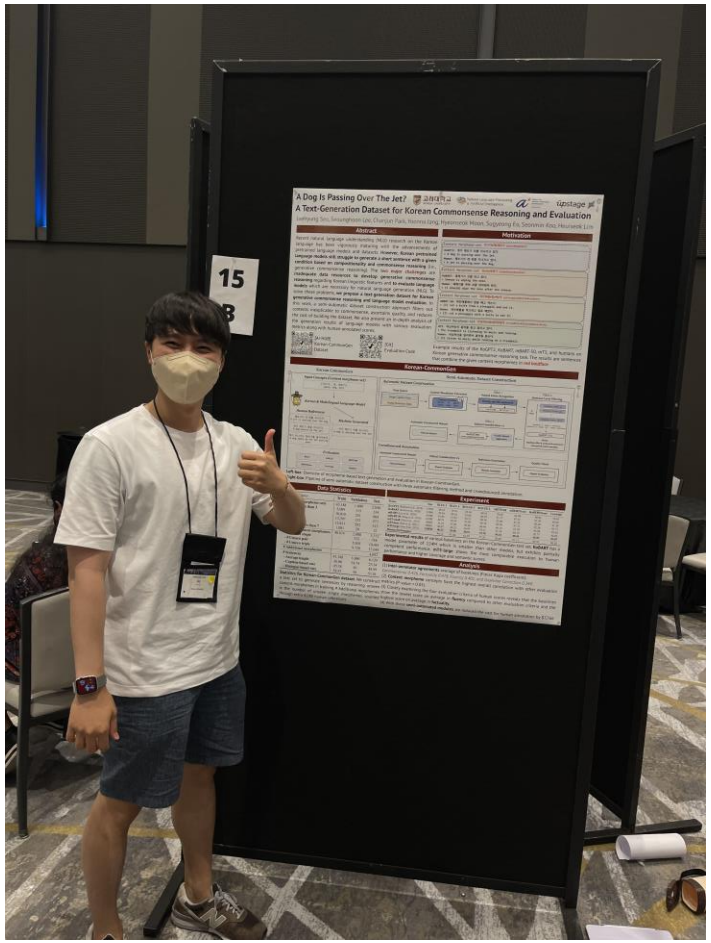
2022 NAACL paper presentation

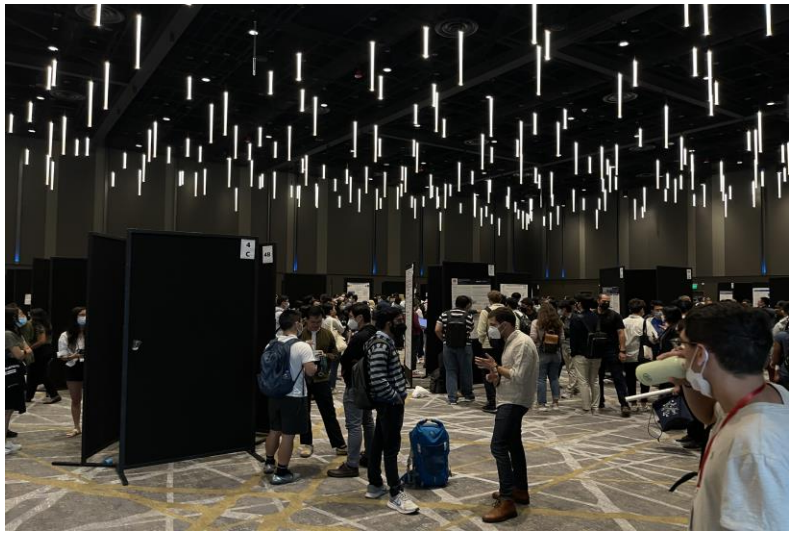
1. Entity Cloze By Date: What LMs Know About Unseen Entities
2. Contrastive Learning for Prompt-Based Few-Shot Language Learners
3. Template-free Prompt Tuning for Few-shot NER

2022. 08. 04

김경민

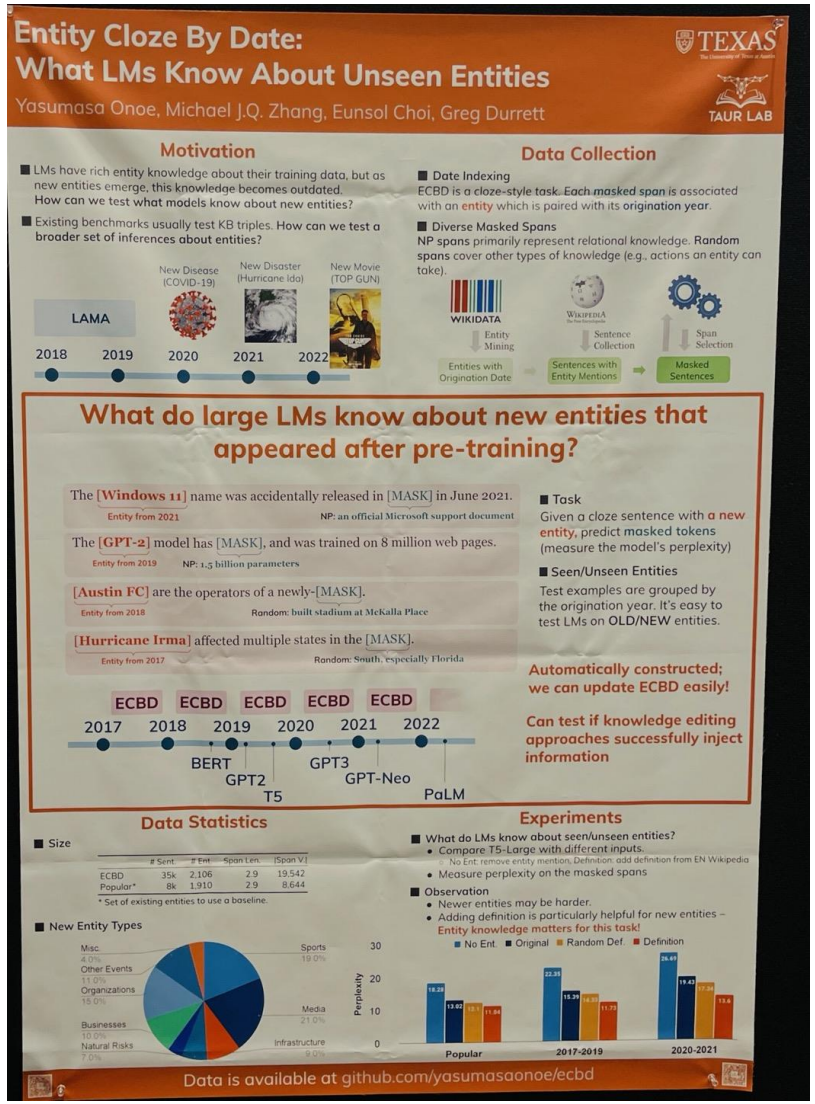
2022 NAACL





2022 NAACL paper presentation - 1

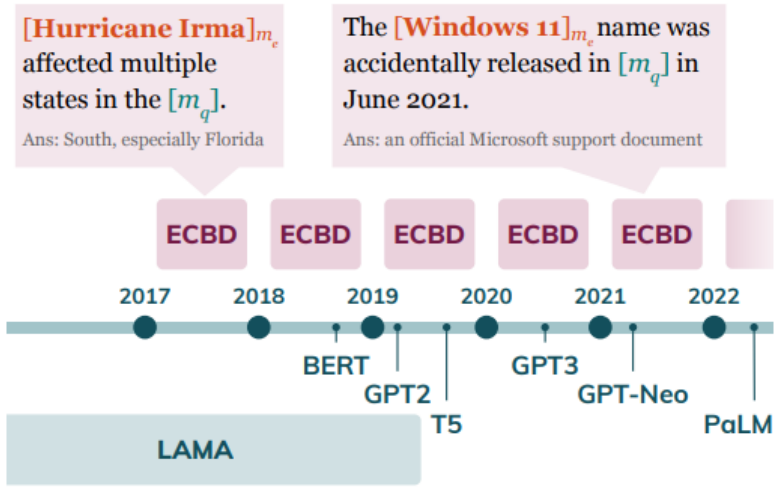
1. Entity Cloze By Date: What LMs Know About Unseen Entities



Entity Cloze By Date: What LMs Know About Unseen Entities

Motivation

- Language models (LMs) are typically trained once on a large-scale corpus and used for years without being updated
- However, in a dynamic world, new entities constantly arise
- We propose a framework to analyze what LMs can infer about new entities that did not exist when the LMs were pretrained
- Existing benchmarks usually test KB triples. How can we test a broader set of inferences about entities?



Entity Cloze By Date: What LMs Know About Unseen Entities

Entity Cloze by Date (ECBD)

1) Test broader entity knowledge
2) Test ability to reason about completely unseen entities

Data Collection

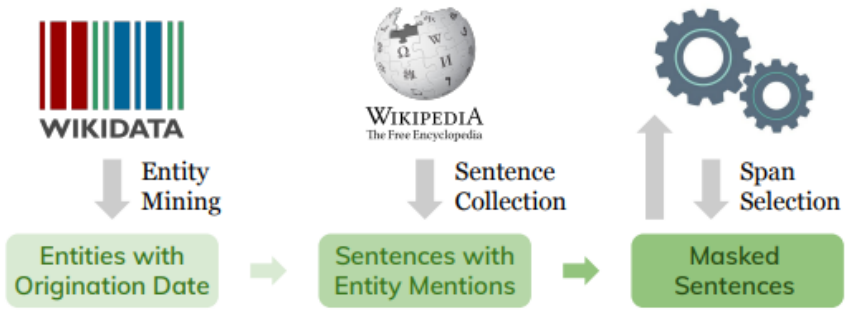


Figure 2: Overview of the data collection process.

Entity Sentence Collection

- Less than 500 words
- Exclude the first paragraph of the article
- Sample sentences that include the entity name or one of their Wikidata aliases
- Do not accept entity mention spans located in quotes
- Filter out any sentences with less than 5 words

Entity Cloze By Date: What LMs Know About Unseen Entities

Task Definition

- Given a cloze sentence with a new entity, predict masked tokens (measure the model's perplexity)

> Each entity e is paired with e_i, its origination year. Given a sentence s containing an entity mention span m_e and a masked query span m_q, a language model is asked to predict the gold masked span m_y. See the following example:
>
> e: RNA vaccine, e_i: 2020
> s: [mRNA vaccines]$_{m_e}$ do not affect or reprogram [m_q].
> m_y: DNA inside the cell
>
> We evaluate language models by **perplexity** on the masked span m_q (see Appendix D for a discussion of recall as another metric).

Seen/Unseen Entities

- Test examples are grouped by the origination year. It's easy to test LMs on OLD/NEW entities

Span Selection

All spans must be:

- (a) not overlapping with the entity mention span, m_e
- (b) located after the entity mention span, m_e
- (c) starting no more than ten words away from the mention span

Extract two types of spans:

- NP spans: suitable noun phrases in the sentence using spaCy
- Random spans: arbitrary sequences of words sampled from the sentence

Entity Cloze By Date: What LMs Know About Unseen Entities

Statistics

Origination Year	2017	2018	2019	2020	2021	Total	Example Entities
# Dev Entities	300	280	219	187	78	1,050	
# Test Entities	299	279	208	176	80	1,029	
Sports	20	19	22	12	27	19	2017 Tour de France, USL League One, Evo 2017
Media	18	19	24	23	20	21	Emily in Paris, Luigi's Mansion 3, The Midnight Gospel
Infrastructure	10	8	10	8	9	9	Gateway Arch National Park, Istanbul Airport, I-74 Bridge
Natural Risks	3	6	4	15	11	7	Hurricane Ida, COVID-19, North Complex Fire
Products	4	4	4	3	3	4	Apple Card, Sputnik V COVID-19 vaccine, Pixel 4
Businesses	15	11	7	7	3	10	Raytheon Technologies, Electrify America, Good Party
Organizations	16	18	13	12	9	15	NUMTOT, UK Student Climate Network
Other Events	9	10	11	12	13	11	Super Bowl LIV halftime show, Storm Area 51
Misc.	5	3	4	7	4	4	RNA vaccine, Earthshot Prize, Comet NEOWISE

Table 1: Origination date indexed entity (ODIE) statistics by category. The number represents % of entities with particular type among entities originated in that year.

Entity Cloze By Date: What LMs Know About Unseen Entities

Experiment Setup

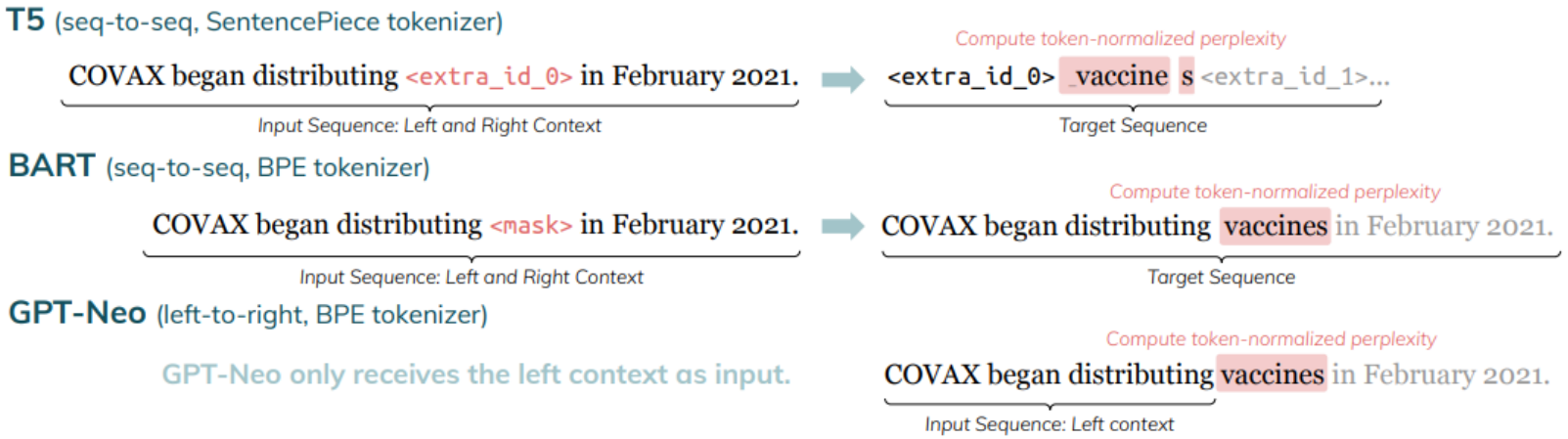


Figure 3: Perplexity computation over the masked span with three different modeling paradigms.

Entity Cloze By Date: What LMs Know About Unseen Entities

Seen entities | Unseen entities

	POPULAR	2017-2019	2020-2021
Type: seq-to-seq	**T5 Large**		Size: 770M
ORIGINAL	13.02	15.39	19.43
NO ENT	18.28	22.35	26.69
RANDOM DEF.	12.10	14.33	17.34
DEFINITION	11.04	11.73	13.60
Δ (ORIG. → RAND.)	-0.92	-1.06	-2.09
Δ (ORIG. → DEF.)	-1.98	-3.66	-5.83
Type: seq-to-seq	**BART Large**		Size: 406M
ORIGINAL	22.70	21.09	28.79
NO ENT	33.33	30.56	39.25
RANDOM DEF.	27.69	25.59	33.74
DEFINITION	21.10	17.66	22.00
Δ (ORIG. → RAND.)	+4.99	+4.50	+4.95
Δ (ORIG. → DEF.)	-1.60	-3.43	-6.79
Type: left-to-right	**GPT-Neo**		Size: 1.3B
ORIGINAL	28.61	27.81	33.36
NO ENT	54.01	51.46	54.81
RANDOM DEF.	39.46	41.03	45.92
DEFINITION	23.19	19.09	22.33
Δ (ORIG. → RAND.)	+10.85	+13.22	+12.56
Δ (ORIG. → DEF.)	-5.42	-8.72	-11.03

Table 3: Results of T5, BART, and GPT-Neo on the test set, showing perplexity (↓).

ORIGINAL: original masked sentence

No ENT: replaces the entity mention span with “the entity”

RANDOM DEF: prepends a definition sentence of a randomly selected entity

DEFINITION: prepends the first sentence of the entity’s Wikipedia article to the cloze sentence

Consistent trends across three LMs

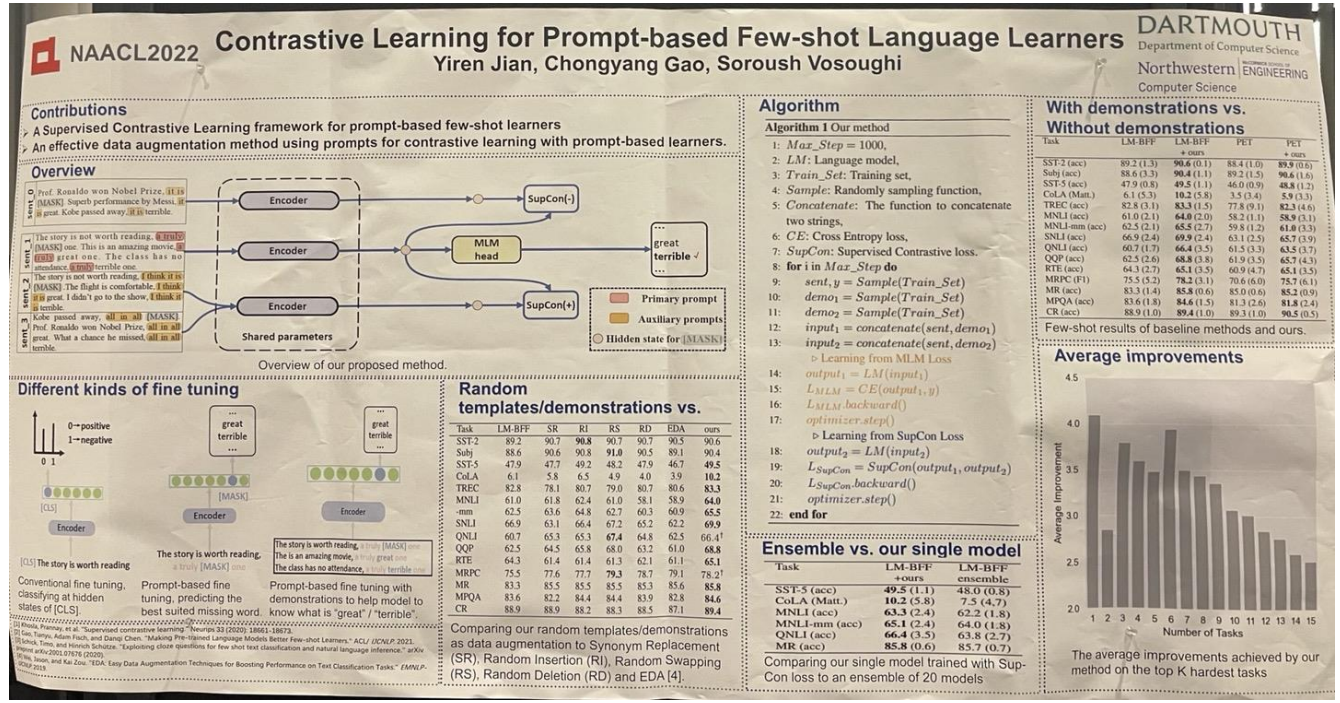
1. No ENT always degrades performances compared to ORIGINAL
   => Our masked spans are sensitive

2. DEFINITION always boosts performance over ORIGINAL
   => provide more information about entities helps to retrieve information distribted over LM’s parametersOur masked spans are sensitive

Conclusion

- We present a dataset to understand language models' broad inferences about entities across time
- We collect 43k cloze-style sentences associated with a time-indexed set of entities
- We also perform analysis on our dataset and show that handling completely unseen entities remains challenging for the current LMs

Contrastive Learning for Prompt-Based Few-Shot Language Learners



Contrastive Learning for Prompt-Based Few-Shot Language Learners

Motivation

- The impressive performance of GPT-3 using natural language prompts and in-context learning has inspired work on better fine-tuning of moderately-sized models under this paradigm
- Following this line of work, we present a contrastive learning framework that clusters inputs from the same class for better generality of models trained with only limited examples
- Supervised contrastive framework that clusters inputs from the same class under different augmented "views" and repel the ones from different classes

- Main contribution
  - A Supervised Contrastive Learning framework for prompt-based few-shot learners
  - An effective data augmentation method using prompts for contrastive learning with prompt-based learners

Supervised Contrastive Learning (SupCon)[1)]

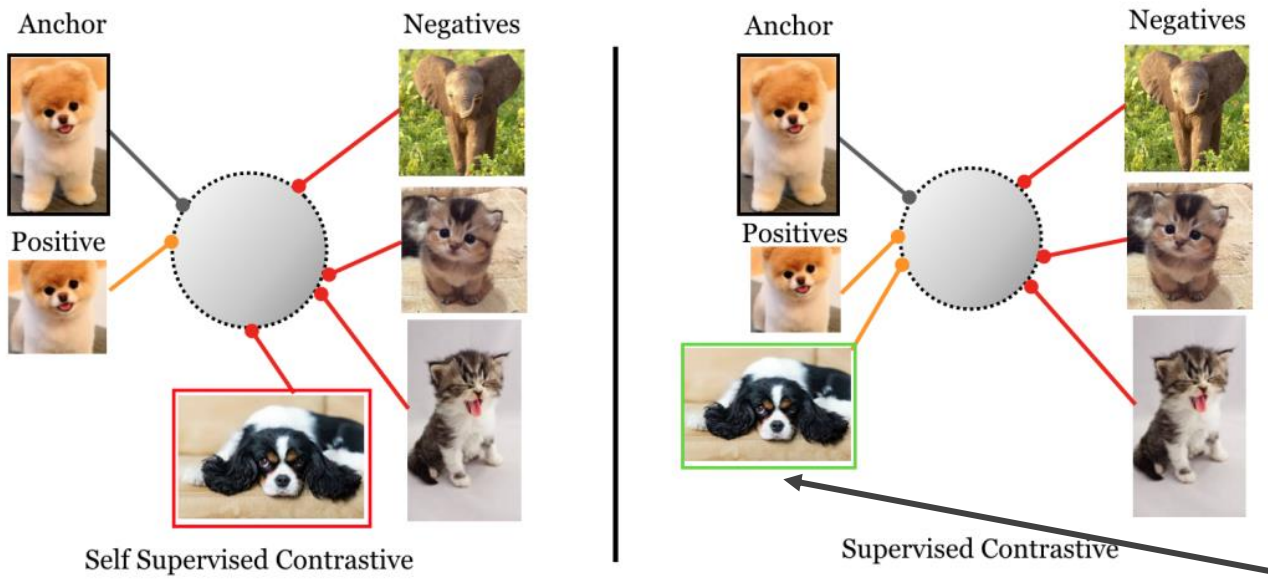


Figure 2: Supervised vs. self-supervised contrastive losses: The self-supervised contrastive loss (left, Eq. 1) contrasts a *single* positive for each anchor (i.e., an augmented version of the same image) against a set of negatives consisting of the entire remainder of the batch. The supervised contrastive loss (right) considered in this paper (Eq. 2), however, contrasts the set of *all* samples from the same class as positives against the negatives from the remainder of the batch. As demonstrated by the photo of the black and white puppy, taking class label information into account results in an embedding space where elements of the same class are more closely aligned than in the self-supervised case.

Different augmented "views" (augmentation)

1) https://github.com/google-research/google-research/tree/master/supcon

Method

(sent) (temp[mask]) (demo)

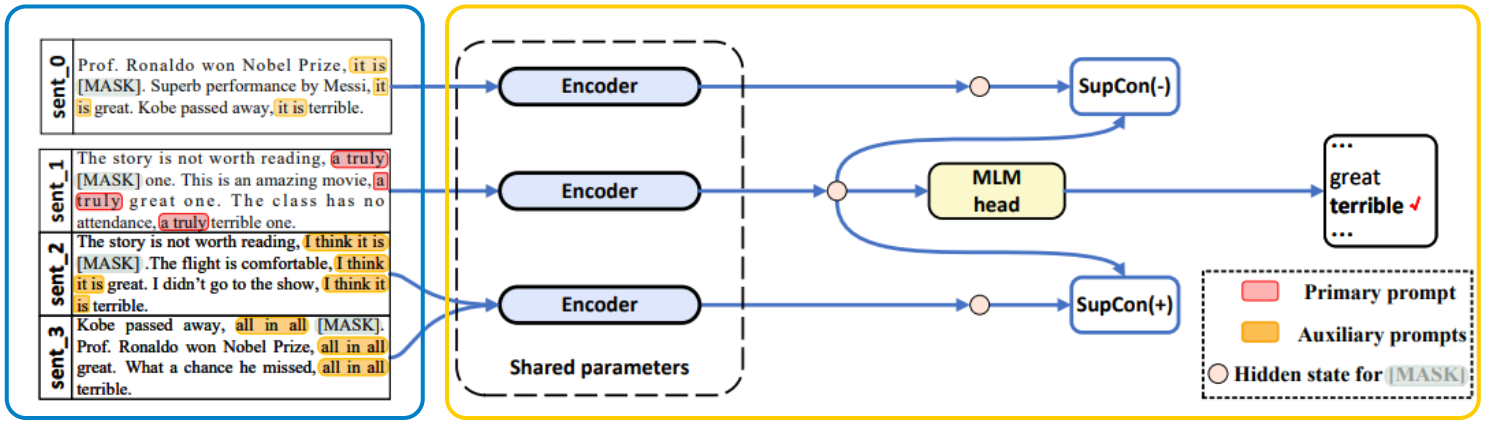


Figure 1: Overview of our proposed method. Besides the standard prompt-base MLM loss on label words "great" and "terrible", we introduce a SupCon loss on multi-views of input text. The positive pair is sentences (with sampled templates and/or demonstrations) in the same class, e.g. sent_1 and sent_3, or itself with a different template and demonstrations, e.g. sent_1 and sent_2. The negative sentence pair is input sentences (with sampled templates and/or demonstrations) in different classes, e.g. sent_1 and sent_0.

Algorithm 1 Our method

$Max\_Step = 1000$,
LM: Language model,
$Train\_Set$: Training set,
$Sample$: Randomly sampling function,
$Concatenate$: The function to concatenate two strings,
CE: Cross Entropy loss,
$SupCon$: Supervised Contrastive loss.
for i in Max_Step **do**
  $sent, y = Sample(Train\_Set)$
  $demo_1 = Sample(Train\_Set)$
  $demo_2 = Sample(Train\_Set)$
  $input_1 = concatenate(sent, demo_1)$
  $input_2 = concatenate(sent, demo_2)$
  ▷ Learning from MLM Loss
  $output_1 = LM(input_1)$
  $L_{MLM} = CE(output_1, y)$
  $L_{MLM}.backward()$
  $optimizer.step()$
  ▷ Learning from SupCon Loss
  $output_2 = LM(input_2)$
  $L_{SupCon} = SupCon(output_1, output_2)$
  $L_{SupCon}.backward()$
  $optimizer.step()$
end for

Experiment

Task	LM-BFF	LM-BFF +Dec	LM-BFF +Dec +Lab	LM-BFF +ConCal	LM-BFF +ours
SST-2	89.2 (1.3)	90.1 (0.6)	**90.6** (0.5)	88.5 (2.0)	**90.6** (0.1)
Subj	88.6 (3.3)	87.3 (3.6)	88.4 (4.9)	83.8 (7.3)	**90.4** (1.1)
SST-5	47.9 (0.8)	47.2 (1.0)	46.5 (0.7)	47.9 (1.1)	**49.5** (1.1)
CoLA	6.1 (5.3)	9.8 (6.5)	7.2 (5.2)	6.7 (4.6)	**10.2** (5.8)
TREC	82.8 (3.1)	81.9 (3.0)	82.3 (3.0)	71.1 (7.0)	**83.3** (1.5)
MNLI	61.0 (2.1)	61.3 (2.1)	59.4 (1.3)	61.0 (0.8)	**64.0** (2.0)
-mm	62.5 (2.1)	63.2 (2.1)	61.4 (1.6)	62.5 (0.8)	**65.5** (2.7)
SNLI	66.9 (2.4)	67.0 (3.1)	65.8 (2.1)	67.0 (2.9)	**69.9** (2.4)
QNLI	60.7 (1.7)	60.0 (2.5)	60.2 (2.0)	60.9 (2.0)	**66.4** (3.5)
QQP	62.5 (2.6)	**69.0** (1.7)	65.4 (1.2)	62.2 (2.7)	68.8 (3.8)
RTE	64.3 (2.7)	**65.6** (1.5)	65.3 (2.4)	60.2 (1.9)	65.1 (3.5)
MRPC	75.5 (5.2)	69.4 (7.0)	66.5 (7.0)	**78.3** (3.1)	$78.2^{\dagger}$ (3.1)
MR	83.3 (1.4)	85.0 (1.0)	84.6 (1.2)	84.0 (1.4)	**85.8** (0.6)
MPQA	83.6 (1.8)	82.3 (1.9)	84.3 (1.4)	72.3 (13.4)	**84.6** (1.5)
CR	88.9 (1.0)	89.3 (0.6)	**89.6** (0.7)	87.7 (1.1)	89.4 (1.0)

Table 2: Comparing our SupCon loss with Decoupling Label Loss (Dec), Label Condition Loss (Lab), and Contextual Calibration (ConCal). † We can achieve stronger performance 80.0 ± 1.8 by fixing templates/demonstrations when creating the second view of the input (see Section 6.2).

Task	LM-BFF	- demo + temp	+ demo - temp	+ demo + temp
SST-2 (acc)	89.2 (1.3)	**90.8** (0.3)	90.5 (0.4)	90.6 (0.1)
Subj (acc)	88.6 (3.3)	**90.8** (0.8)	90.6 (1.2)	90.4 (1.1)
SST-5 (acc)	47.9 (0.8)	49.3 (1.7)	48.9 (1.8)	**49.5** (1.1)
CoLA (Matt.)	6.1 (5.3)	9.9 (7.5)	8.5 (5.6)	**10.2** (5.8)
TREC (acc)	82.8 (3.1)	83.4 (0.5)	**86.7** (1.0)	83.3 (1.5)
MNLI (acc)	61.0 (2.1)	63.4 (3.3)	63.0 (3.2)	**64.0** (2.0)
MNLI-mm (acc)	62.5 (2.1)	65.5 (3.1)	64.9 (3.4)	**65.5** (2.7)
SNLI (acc)	66.9 (2.4)	69.8 (2.4)	68.5 (1.9)	**69.9** (2.4)
QNLI (acc)	60.7 (1.7)	65.4 (3.1)	**67.0** (3.6)	66.4 (3.5)
QQP (acc)	62.5 (2.6)	**68.9** (3.2)	67.8 (1.4)	68.8 (3.8)
RTE (acc)	64.3 (2.7)	64.9 (3.8)	62.6 (2.8)	**65.1** (3.5)
MRPC (F1)	75.5 (5.2)	79.0 (1.8)	**80.0** (1.8)	78.2 (3.1)
MR (acc)	83.3 (1.4)	85.8 (0.7)	85.4 (0.3)	**85.8** (0.6)
MPQA (acc)	83.6 (1.8)	84.0 (1.9)	84.1 (2.0)	**84.6** (1.5)
CR (acc)	88.9 (1.0)	88.6 (0.6)	88.2 (1.0)	**89.4** (1.0)

Table 5: Different strategies to construct multi-views of input sentences. Fixed demonstrations and sampling templates (- demo + temp), sampling demonstrations and fixed templates (+ demo - temp) and sampling both demonstrations and templates (+ demo + temp).

Conclusion

- We proposed a novel supervised contrastive learning framework and an effective augmentation method using prompts that can boost the performance of prompt-based language learners and outperform recent work on 15 few-shot tasks

Template-free Prompt Tuning for Few-shot NER

Template-free Prompt Tuning for Few-shot NER

Motivation

- Prompt-based methods have been successfully applied in sentence-level few-shot learning tasks, mostly owing to the sophisticated design of templates and label words
- In NER, it would be time-consuming to enumerate the template queries over all potential entity spans
- Propose a more elegant method to reformulate NER tasks as LM problems without any templates
- Discard the template construction process while maintaining the word prediction paradigm of pre-training models to predict a class-related pivot word (or label word) at the entity position

- Main contribution
  - Propose a template-free approach to prompt NER under few-shot setting
  - explore several approaches for label word engineering accompanied with intensive experiments
  - Experimental results verify the effectiveness of the proposed method under few-shot setting. Meanwhile, the decoding speed of the proposed method is 1930.12 times faster than template-based baseline

Template-free Prompt Tuning for Few-shot NER

Challenges in NER

1. Searching for appropriate templates is harder as the search space grows larger when encountering span-level querying in NER. What's worse, such searching with only a few annotated samples as guidance can easily lead to overfitting

2. Obtaining the label of each token requires enumerating all possible spans, which would be time-consuming

Figure 1: An example of template-based prompt method for NER. Predicting all labels in sentence "Obama was born in America." requires enumeration over all spans.

NER as an LM task
⇒ Entity-oriented LM (EntLM) objective

Method

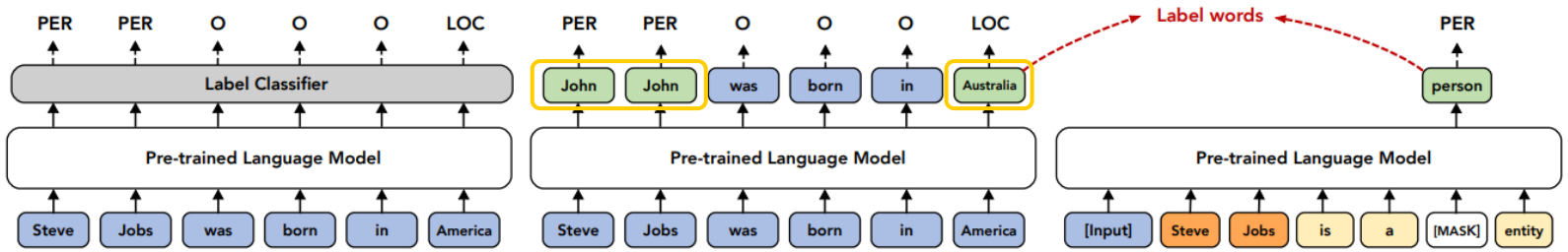


Figure 2: Comparison of different fine-tuning methods for NER. (a) is the standard fine-tuning method, which replace the LM head with a classifier head and perform label classification. (c) is the template-based prompt learning method, which induces the LM to predict label words by constructing a template. (b) is the proposed Entity-oriented LM fine-tuning method, which also re-uses the LM head and leads the LM to predict label words through an Entity-oriented LM objective. (For entities with multiple spans, the model predicts the same label word at each position, which is similar to the “IO" labeling scheme.)

Experiment

Datasets	Domain	# Class	# Train	# Test
CoNLL'03	News	4	14.0k	3.5k
OntoNotes*	General	11	60.0k	8.3k
MIT Movie	Review	12	7.8k	2.0k

Table 1: Dataset details. OntoNotes* denotes the Ontonotes5.0 dataset after removing value/numerical/time/date entity types.

Datasets	Methods	K=5	K=10	K=20	K=50
CoNLL03	BERT-tagger (IO)	41.87 (12.12)	59.91 (10.65)	68.66 (5.13)	73.20 (3.09)
	NNShot	42.31 (8.92)	59.24 (11.71)	66.89 (6.09)	72.63 (3.42)
	StructShot	45.82 (10.30)	62.37 (10.96)	69.51 (6.46)	74.73 (3.06)
	Template NER	43.04 (6.15)	57.86 (5.68)	66.38 (6.09)	72.71 (2.13)
	EntLM (Ours)	49.59 (8.30)	64.79 (3.86)	69.52 (4.48)	73.66 (2.06)
	EntLM + Struct (Ours)	**51.32** (7.67)	**66.86** (3.01)	**71.23** (3.91)	**74.80** (1.87)
OntoNotes 5.0	BERT-tagger (IO)	34.77 (7.16)	54.47 (8.31)	60.21 (3.89)	68.37 (1.72)
	NNShot	34.52 (7.85)	55.57 (9.20)	59.59 (4.20)	68.27 (1.54)
	StructShot	36.46 (8.54)	57.15 (5.84)	62.22 (5.10)	68.31 (5.72)
	Template NER	40.52 (8.62)	49.89 (3.66)	59.53 (2.25)	65.15 (2.95)
	EntLM (Ours)	45.21 (9.17)	57.64 (4.18)	65.64 (4.24)	71.77 (1.31)
	EntLM + Struct (Ours)	**46.60** (10.35)	**59.35** (3.24)	**67.91** (4.55)	**73.52** (0.97)
MIT-Movie	BERT-tagger (IO)	39.57 (6.38)	50.60 (7.29)	59.34 (3.66)	71.33 (3.04)
	NNShot	38.97 (5.54)	50.47 (6.09)	58.94 (3.47)	71.17 (2.85)
	StructShot	41.60 (8.97)	53.19 (5.52)	61.42 (2.98)	72.07 (6.41)
	Template NER	45.97 (3.86)	49.30 (3.35)	59.09 (0.35)	65.13 (0.17)
	EntLM (Ours)	46.62 (9.46)	57.31 (3.72)	62.36 (4.14)	71.93 (1.68)
	EntLM + Struct (Ours)	**49.15** (8.91)	**59.21** (3.96)	**63.85** (3.7)	**72.99** (1.80)

Table 2: Main results of EntLM on three datasets under different few-shot settings (K=5,10,20,50). We report mean (and deviation in brackets) performance over 3 different splits (4 repeated experiments for each split).

Conclusion

- Propose a template-free prompt tuning method, EntLM, for few-shot NER
- Reformulate the NER task as an Entity-oriented LM task, which induce the LM to predict label words at entity positions during fine-tuning
- Experimental results show that the proposed method can achieve significant improvement on few-shot NER over BERT-tagger and template-based method
- Decoding speed of EntLM is up to 1930.12 times faster than the template-based method

THANK YOU

Q&A