Beyond pre-trained models: how inserting knowledge can improve commonsense reasoning

Caterina Lacerra
lacerra@di.uniroma1.it
Sapienza NLP - Sapienza, University of Rome

Diverse and Informative Dialogue Generation with Context-Specific Commonsense Knowledge Awareness

Sixing Wu, Ying Li, Dawei Zhang, Yang Zhou and Zhonghai Wu

Peking University, Beijing, China

Auburn University, Alabama, USA

https://github.com/pku-orangecat/ACL2020-ConKADI

Open-domain Dialogue Generation

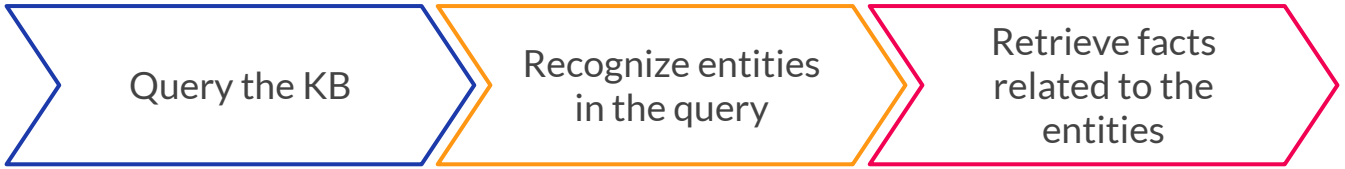
Given a query message, the dialogue system has to generate a response.

Traditional generative open-domain dialogue systems tend to produce **generic** answers, such as *"I don't know"*.

Recent approaches introduced large-scale **knowledge graphs** to enhance dialogue generation.

Open-domain Dialogue Generation

Given a query message X and a set of **common sense facts** F, the dialogue system has to generate a response Y.



Disambiguate the entity.

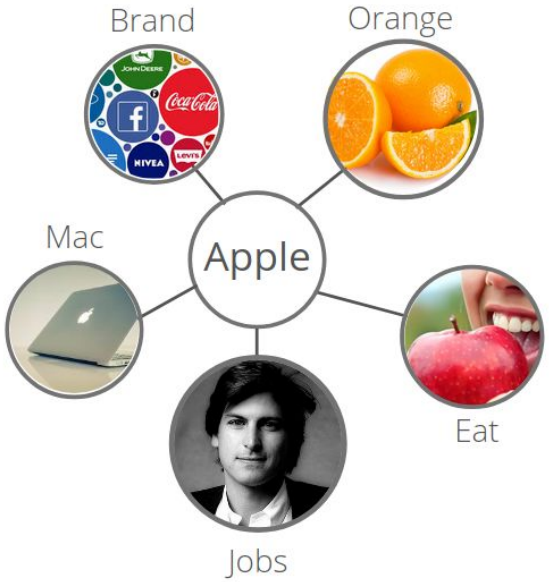
Facts must be **coherent** with the dialogue.

Knowledge should be **integrated** properly.

Knowledge Retriever

Facts are retrieved from a knowledge graph G, starting from the query X.

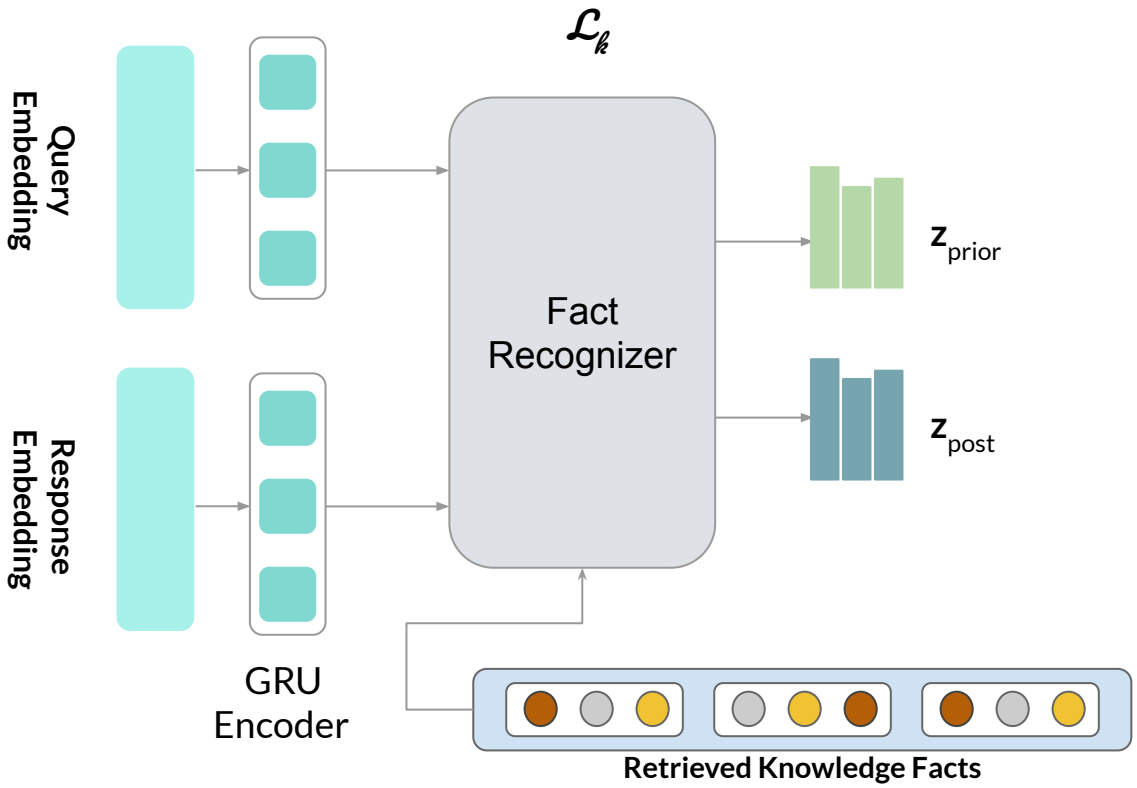
`X: Apple's new product is awesome.`



If the entity can be matched to a vertex *v* in *G*, then retrieve as candidate each neighbour of *v* and the respective relations:

(apple, isA, brand)
(apple, canBe, eaten)
...
(apple, relatedTo, orange)

Fact Recognizer



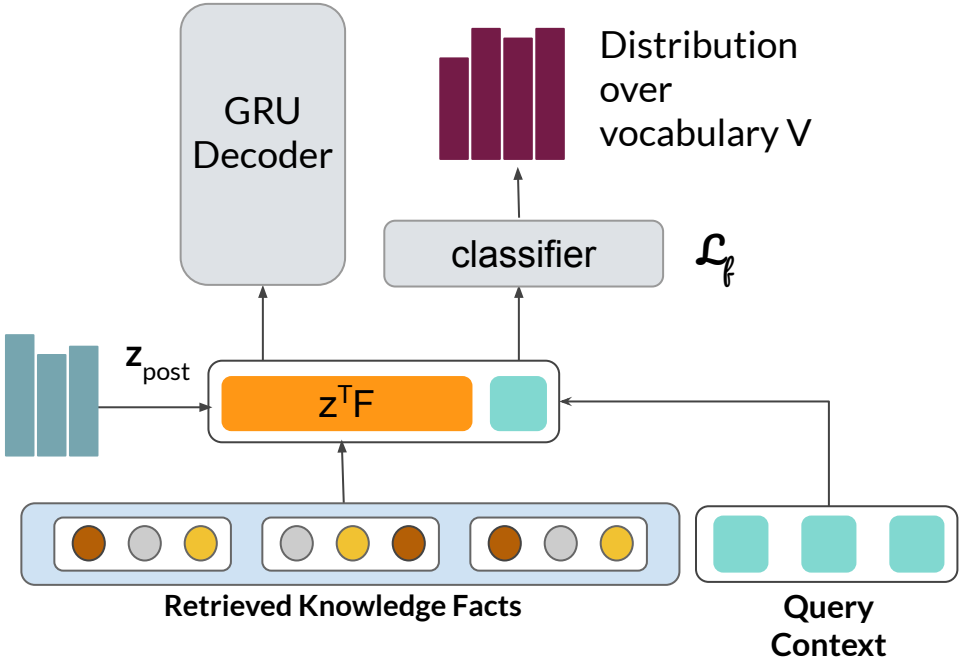
Learn to recognize facts that are related to the context:

Fact recognizer module (dense layer), fed with the contextualized representation of the input, outputs a distribution **z** over the retrieved facts.

The contextualized representation is obtained with a GRU encoder.

At training time, the responses are fed to the network and the divergence between z_{post} and z_{prior} is minimized.

Context-Knowledge Fusion



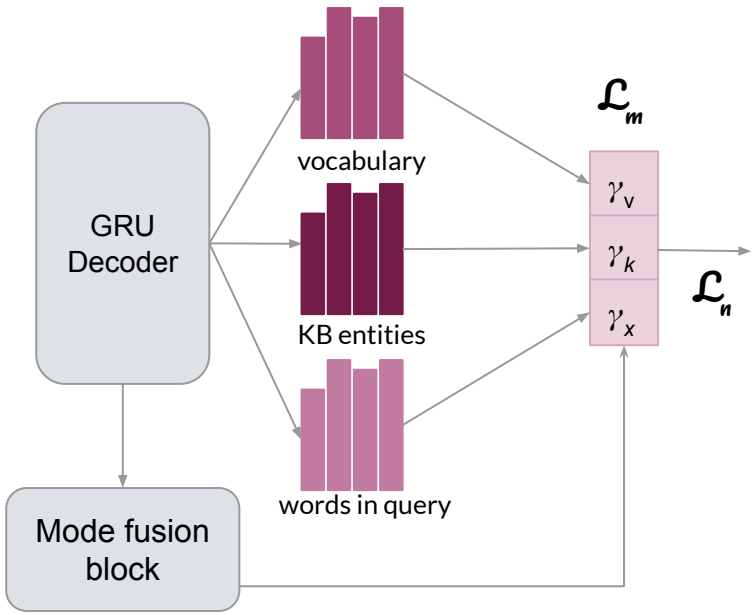
Put context and knowledge together:

The concatenation of the **weighted representation of facts** and **query context** feeds a classifier and a GRU decoder.

The loss $\mathcal{L}_f$ is the difference of two components:

- One takes into account z_{post} and a binary indicator (1 if fact f is in Y).
- One is computed on the distribution over vocabulary V.

Triple Knowledge Detector



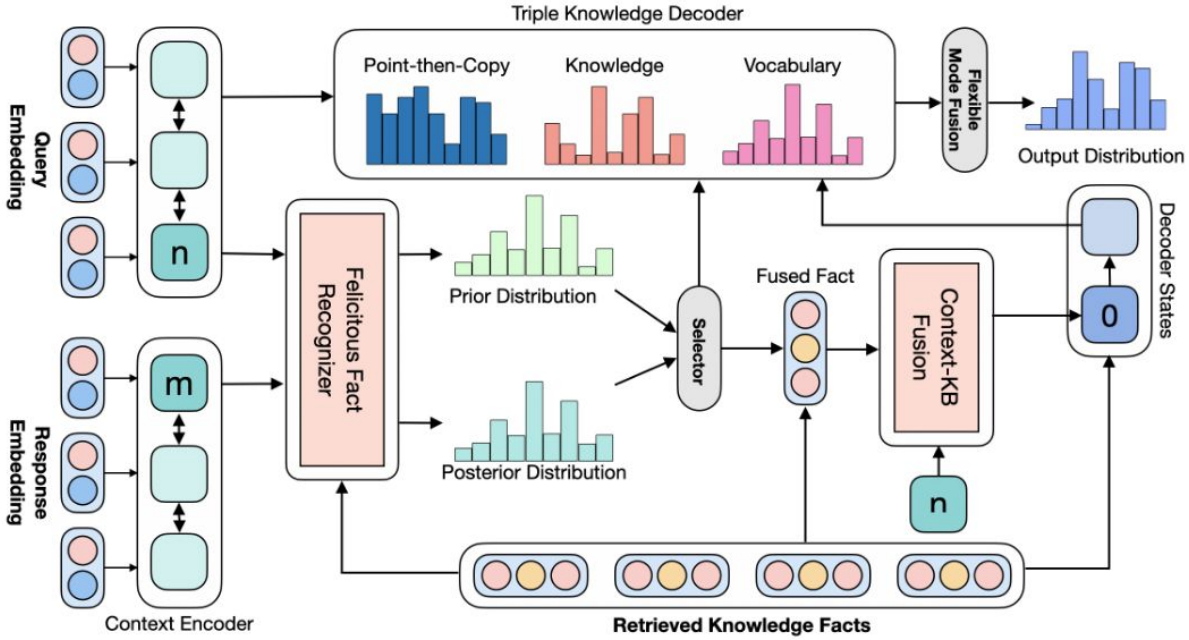
Find a balance among copying input, generating word, and using KB:

A *mode fusion block* outputs a distribution over the three available *modes,* taking as input the decoder's hidden state and its previous time step.

CE loss $\mathcal{L}_m$ is computed between the ground-truth mode and the predicted distribution.

The final output distribution is optimized taking into account both the difference between the output and the gold, and the loss $\mathcal{L}_m$.

ConKADI - The whole model



$$\mathcal{L} = \mathcal{L}_k + \mathcal{L}_f + \mathcal{L}_n$$

Experiments - Metrics

E_{match} : averaged number of the matched target entities per generated text.

E_{recall} : recalled entities in the generated text.

BLEU-*n*: overlap of *n*-grams with the gold standard.

Distinct-*n*: ratio of distinct *n*-grams in all generated texts.

Datasets

Reddit (English): subset of CCM Reddit Commonsense Dataset.

Weibo (Chinese): built upon three previous open Chinese Weibo datasets, with entities linked to ConceptNet.

Competitors

S2S (Sutskever et al., 2014**):** Sequence to Sequence architecture**.**

ATS2S$_{MMI}$(Li et al., 2016a**):** Attention + S2S + bidi-MMI.

Copy (Gu et al., 2016; Vinyals et al., 2015): Decoder points and copies from the input query.

ConKADI$_{-cp}$: copying from the input query is not allowed.

Experiments - Results

Chinese Weibo Dataset - Dialogue Generation Task

System	E_{match}	E_{recall}	BLEU-2	BLEU-3	Distinct-1	Distinct-2
S2S	0.33	0.13	2.24	0.80	0.21	1.04
$ATS2S_{MMI}$	0.40	0.15	4.01	**1.61**	0.75	3.91
Copy	0.33	0.13	2.28	0.84	0.59	2.18
ConKADI	1.48	**0.38**	**5.06**	1.59	**3.26**	**23.93**
$ConKADI_{-cp}$	**1.60**	**0.38**	5.00	1.52	2.34	18.29

Experiments - Results

English Reddit Dataset - Dialogue Generation Task

System	E_{match}	E_{recall}	BLEU-2	BLEU-3	Distinct-1	Distinct-2
S2S	0.41	0.04	4.81	1.89	0.38	1.77
ATS2S	0.44	0.05	4.50	1.81	0.82	3.44
Copy	0.13	0.09	**5.43**	**2.26**	1.73	8.33
ConKADI	1.24	**0.14**	3.53	1.27	**2.77**	**18.78**
$ConKADI_{-cp}$	**1.60**	**0.13**	3.09	1.07	2.29	16.70

Experiments - Human Annotation

Three human annotators were asked to evaluate a sample of the generated answers against the gold ones in terms of:

- Appropriateness, that includes both *fluency* and *logical relevance*.
- Informativeness, i.e. whether the answer provides new information/knowledge or not.

Experiments - Human Annotation

	Appropriateness			Informativeness		
	Win	**Tie**	**Lose**	**Win**	**Tie**	**Lose**
ConKADI vs. S2S	0.71	0.11	0.17	0.87	0.07	0.06
ConKADI vs. $ATS2S_{MMI}$	0.59	0.09	0.31	0.82	0.07	0.10
ConKADI vs. Copy	0.72	0.08	0.19	0.90	0.04	0.06

Chinese Weibo Dataset

Experiments - Case Study

	English Reddit Dataset	**Chinese Weibo Dataset**
Query	My cat likes bananas and bread.	I can sing the song of Taylor Swift.
ATS2S	I'm a banana and I don't know what you're talking about.	Me too, I'm singing **unk.**
$ATS2S_{MMI}$	Do you have a cat?	What kind of song are you singing?
Copy	I'm a cat.	Me too, me too, me too, I like it.
ConKADI	And your cat is the best.	I heard it. His music is good.

Conclusions

- New knowledge-aware model that is trained to retrieve facts that are relevant to the context, merging them in a proper way to the dialogue.
- SOTA results in many evaluation tests, both for English and Chinese languages.
- Human evaluation of generated dialogue for Chinese outperform other competitors.

Knowledge-Enhanced Pretraining Model for Commonsense Story Generation

Jian Guan, Fei Huang, Zhihao Zhao, Xiaoyan Zhu and Minlie Huang

https://github.com/thu-coai/CommonsenseStoryGen

http://coai.cs.tsinghua.edu.cn/static/CommonsenseStoryGen/

Story Generation

From an initial sentence, generate *k* more sentences that form a **coherent** and **logical** story.

Usually solved with **generative models**, that have shown good results, but still suffer from repetitions, logic conflicts and lack of long-range coherence.

Story Generation: an example

The first time I saw the results of an **accident**, I was in a bus.

Golden Continuation by human:

We were stopped and a friend pointed out the accident on the road. A man was under a car after having been run over. People tried to move the car off him but only made it worse. After seeing that, I am much more careful when crossing the street.

Pretrained GPT-2:

She was trying to get home but the other passengers would not let her go. He thought she was going to die because of her weight. She was trying to get home but the other passengers would not let her go. The bus driver **would not let her go.**

Fine-tuned GPT-2:

I was on my way to a party. I was on my way to a party and I'd gotten out of my seat, and started driving. I got a flat tire, so **I stopped driving. I drove** to the party and had a great time.

- ✓ Good fluency
- ✓ Intra-sentence coherence
- ✗ Ignoring **key** entities from the context
- ✗ **Repetitions**
- ✗ **Conflicting logic**

Story Generation: proposed solution

Add commonsense knowledge to a generative model:

- **Retrieve** concepts that are related to the entities in the input context from an external commonsense knowledge base.
- **Fine-tune** GPT-2 on examples from the retrieved concepts.

Generate reasonable stories:

- Add **auxiliary classification task** to distinguish between true and fake stories, while fine-tuning on a story generation dataset.

Training with commonsense knowledge

Extract triples from existing commonsense knowledge bases and transform them into natural language sentences. Fine-tune GPT-2 on the obtained dataset.

Knowledge Bases	Original Triples	Transformed Sentences
ConceptNet	(eiffel tower, **AtLocation**, paris) (telephone, **UsedFor**, communication)	eiffel tower **is at** paris. telephone **is used for** communication.
ATOMIC	(PersonX cooks spaghetti, **xIntent**, to eat)	PersonX cooks spaghetti. PersonX **wants to** eat.
	(PersonX dates for years, **oEffect**, continue dating)	PersonX dates for years. PersonY **will** continue dating.

Training with commonsense knowledge

Fine-tune GPT-2 on the new created dataset:

Dataset	Training	Validation	Test
ROCStories	88,344	4,908	4,909
ConceptNet	600,000	2,400	2,400
ATOMIC	574,267	70,683	64,456

Multi-task Learning

Add an auxiliary classification task, while fine-tuning GPT-2 on ROCStories dataset.

This dataset is a corpus of five-sentence commonsense stories, with a focus on daily-events causal and temporal relations.

Multi-task Learning - fake datasets construction

D1: ROCStories

0. [FEMALE] has an exam tomorrow.
1. She got so stressed, she pulled an all-nighter.
2. She went into class the next day.
3. Her teacher stated that the test is postponed for next week.
4. Jennifer felt bittersweet about it.

D2: Shuffled

0. [FEMALE] has an exam tomorrow.
2. She went into class the next day.
1. She got so stressed, she pulled an all-nighter.
4. Jennifer felt bittersweet about it.
3. Her teacher stated that the test is postponed for next week.

D3: Replaced

[FEMALE] has an exam tomorrow.
She got so stressed, she pulled an all-nighter.
She went into class the next day .
She ate a whole plate and said it was good.
Jennifer felt bittersweet about it.

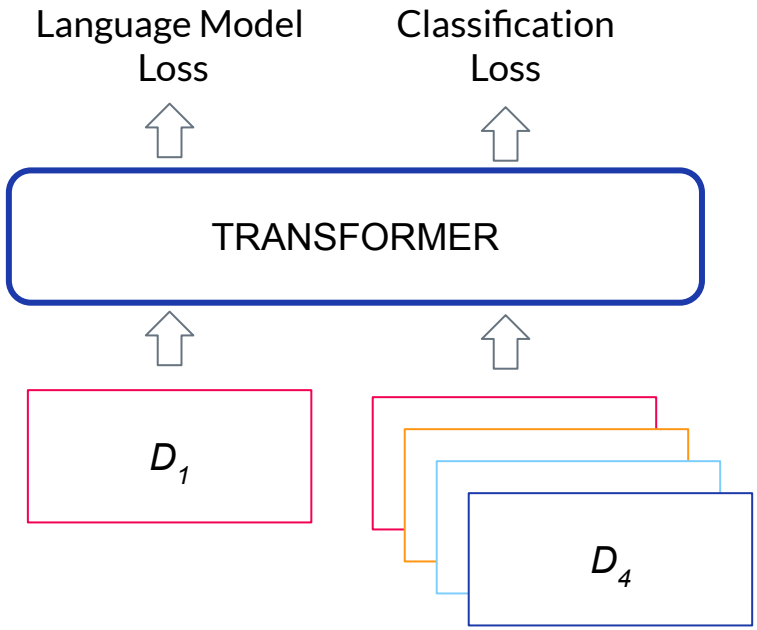
D4: Repetition

[FEMALE] has an exam tomorrow.
She got so stressed, she pulled an all-nighter.
She went into class the next day **She went into class the next day.**
Her teacher stated that the test is postponed for next week.
Jennifer felt bittersweet about it.

Multi-task Learning

The task consists in distinguishing true, coherent stories from fake ones, with the aim of helping the model generate better sentences.

The loss is a weighted loss of the LM and the classification ones.



Multi-task Learning

$$\mathcal{L}_{ST} = \mathcal{L}_{LM} + \lambda \mathcal{L}_{CLS}, \tag{7}$$

$$\mathcal{L}_{LM} = -\sum_{t=1}^{|s|} \log P(s_t | s_{<t}), s \in D_1, \tag{8}$$

$$\mathcal{L}_{CLS} = -\log P(l_s = \tilde{l}_s | s), s \in D_1, D_2, D_3, D_4, \tag{9}$$

where s is a story containing $|s|$ tokens, s_t is the t-th token of s, $\mathcal{L}_{LM}$ is the language modeling loss, $\mathcal{L}_{CLS}$ is the classification loss, and $\tilde{l}_s$ indicates the correct D_i which the story s is sampled from. λ is an adjustable scale factor.

Experiments

Different configurations:

- **GPT-2 Scratch**: training of GPT-2 on ROCStories alone, without pretraining.
- **GPT-2 Pretrain**: pretrained GPT-2. The LM is conditioned on a context of example stories.
- **GPT-2 Fine-tuning**: fine-tuning on ROCStories corpus.

Automatic Evaluation - Metrics

BLEU-*n*: measure of the *n*-gram overlap between human and generated story.

Coverage: average number of commonsense triples matched in each generated story.

Repetition-*n*: percentage of generated stories that repeat at least one *n*-gram.

Distinct-*n*: the ratio of distinct *n*-grams to all the generated *n*-grams.

Automatic Evaluation - Competitors

Convolutional Seq2Seq (ConvS2S): generates a story conditioned upon the beginning based on a convolutional seq2seq model with decoder self-attention. (Gehring et al. 2017)

Fusion Convolutional Seq2Seq (Fusion): pretrains a seq2seq model to generate stories, then provides a fixed version of the model to the a second clone model with fusion mechanism. (Fan et al. 2018)

Plan&Write: It first generates a sequence of keywords as planning, conditioned upon the input, and then generates a story based on the planned keywords. (Yao et al. 2019).

Automatic Evaluation - Competitors

Skeleton-based Model with Reinforcement Learning (SKRL): the model first generates a skeleton (compressed story) with the key sentences, and then generates a story conditioned upon this skeleton. The skeleton is learned by reinforcement learning. (Xu et al. 2018)

Decomposed Model with Semantic Role Labeling (DSRL): It first generates a SRL structure conditioned upon the beginning of the story, then it generates the story on top of the structure. (Fan et al. 2019)

Automatic Evaluation - Results

Models	PPL	BLEU-1	BLEU-2	Coverage	Repetition-4(%)	Distinct-4(%)
ConvS2S	N/A	0.312	0.132	13.64	22.87	72.78
Fusion	N/A	0.322	0.137	12.02	24.23	72.82
Plan&Write	N/A	0.308	0.126	13.38	17.06	67.20
SKRL	N/A	0.267	0.088	10.82	18.34	69.42
DSRL	N/A	0.293	0.117	10.38	**15.36**	73.08
GPT-2 (Scratch)	11.82	0.311	0.134	10.76	22.87	73.33
GPT-2 (Pretrain)	33.50	0.257	0.085	8.04	39.22	64.99
GPT-2 (Fine-tune)	7.96	0.322	0.141	12.40	29.41	73.85
Ours	7.85	**0.326**	**0.143**	**18.48**	21.93	78.96
w/o Pretrain	11.04	0.316	0.134	16.33	21.52	77.17
w/o Knowledge	**7.70**	0.314	0.136	13.95	25.08	73.24
w/o Multi-task	8.04	0.324	0.140	17.19	24.40	**79.43**
Golden Story	*N/A*	*N/A*	*N/A*	*19.28*	*7.64*	*89.51*

Automatic Evaluation - Classification

Story types	D_1	D_2	D_3	D_4
F1 score	0.80	0.81	0.88	0.98

D1: original
D2: shuffled
D3: negative sampling
D4: repeated

Models	Proportional Distribution (%)			
GPT-2 (Pretrain)	15.83	40.8	39.36	4.01
GPT-2 (Fine-tune)	86.94	9.98	2.93	0.15
Ours	90.12	7.98	1.86	0.04
w/o Knowledge	87.76	9.51	2.67	0.06
w/o Multi-task	88.69	9.07	2.02	0.22

Percentage of stories generated for each type of dataset.

Automatic Evaluation - BR and LR

Beginning Ranking accuracy (BR): sample negative beginnings for a true story, and compute the perplexity for the fake and the true story. If the true story has the lowest perplexity, it is considered as a correct prediction.

Logic Ranking accuracy (LR): for each story, compute four fake versions by switching adjacent pairs of sentences. If the perplexity of the true story is the lowest one, it is considered as a correct prediction.

Automatic Evaluation - Results

Models	BR (%)	LR (%)
GPT-2 (Pretrain)	59.3	44.8
GPT-2 (Fine-tune)	73.4	69.6
Ours	76.2	72.7
w/o Knowledge	74.9	71.5
w/o Multi-task	75.7	70.4

Human Evaluation - Metrics

Starting from the same beginning sentence, human annotators were asked to compare couples of generated stories according to **grammaticality** and **logicality.**

The two metrics were evaluated **independently.**

Each story could receive a preference in the set *win*, *lose*, *tie*.

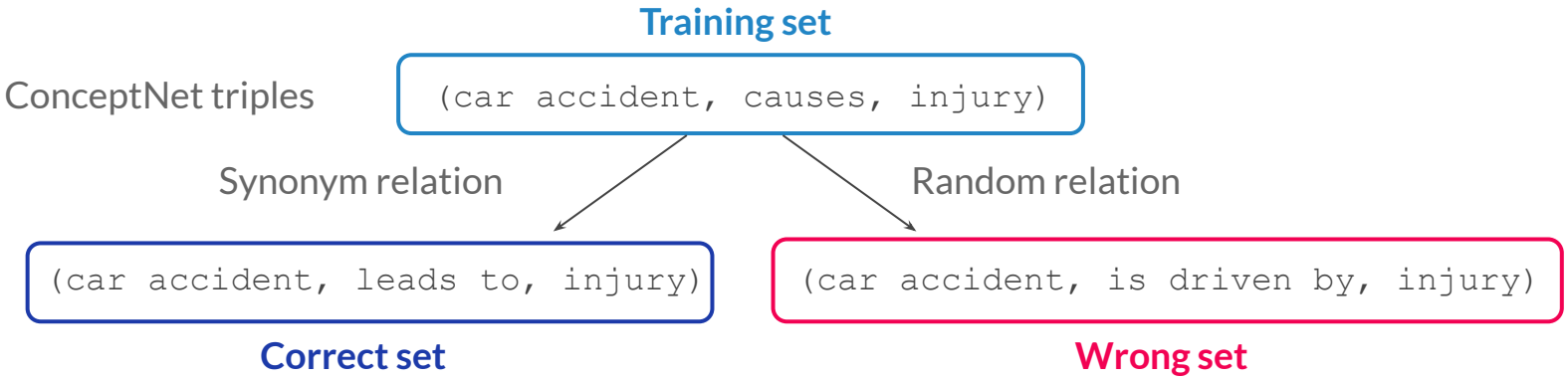
Manual Evaluation - Results

Models	Grammaticality				Logicality			
	Win (%)	Lose (%)	Tie (%)	κ	Win (%)	Lose (%)	Tie (%)	κ
Ours vs. Fusion	50.0**	27.0	23.0	0.421	57.0**	28.0	15.0	0.455
Ours vs. DSRL	58.0**	24.0	18.0	0.441	58.0**	29.0	12.0	0.475
Ours vs. GPT-2 (Scratch)	54.0**	24.5	21.5	0.385	54.0**	26.0	20.0	0.304
Ours vs. GPT-2 (Pretrain)	52.0**	31.5	16.5	0.483	56.5**	32.5	11.0	0.493
Ours vs. GPT-2 (Fine-tune)	42.0**	28.0	30.0	0.344	51.0**	27.5	21.5	0.371
Ours vs. Ours w/o Pretrain	51.0**	31.0	18.0	0.378	56.0**	28.0	16.0	0.375
Ours vs. Ours w/o Knowledge	46.0**	23.0	21.0	0.289	48.0**	29.0	23.0	0.314
Ours vs. Ours w/o Multi-task	37.5	31.0	31.5	0.313	48.5**	25.5	26.0	0.297

Table 5: Manual evaluation results. The scores indicate the percentages of *Win*, *Lose* or *Tie* when our model is compared with a baseline. κ denotes Fleiss' kappa (all are *fair agreement* or *moderate agreement*). The scores marked with * mean p-value< 0.05 and ** indicates p-value< 0.01 in sign test.

Relation Understanding

Does the model capture causal relations from the knowledge base, or it just learns co-occurrences?



Relation Understanding - Results

Measure **accuracy** as percentage of cases where the sentence with wrong template has the highest perplexity compared with the sentences with correct and training templates.

Models	Acc
GPT-2 (Pretrain)	39.28
GPT-2(Fine-tune)	47.48
Ours	67.07
w/o Knowledge	48.07

Case Study

Context: [MALE] was **driving** around in the **snow.**

Golden Story: He needed to **get home** from work. He was driving slowly to **avoid accidents.** Unfortunately the roads were too **slick** and [MALE] **lost control.** His tires **lost traction** and he **hit a tree.**

Generated story**:** Suddenly his car **broke down** on the side of the road. [MALE] had to **call a tow truck.** The tow truck **came and took [MALE] home.** [MALE] was happy he was able to **get home.**

w/o pretrain**:** He tried to stop to see how he could get out of his car. His car started to sputter down on him. [MALE] **pulled over and kept driving.** He was able to make up and the car sped away.

w/o knowledge**:** He noticed a young lady running off. He stopped and picked her up. When he checked his **id** it was lost. [MALE] realized he had forgotten his **id.**

w/o multi-task**:** He noticed a car in the road. He decided to stop. He **got out of his car.** He **drove for half an hour.**

Final Remarks

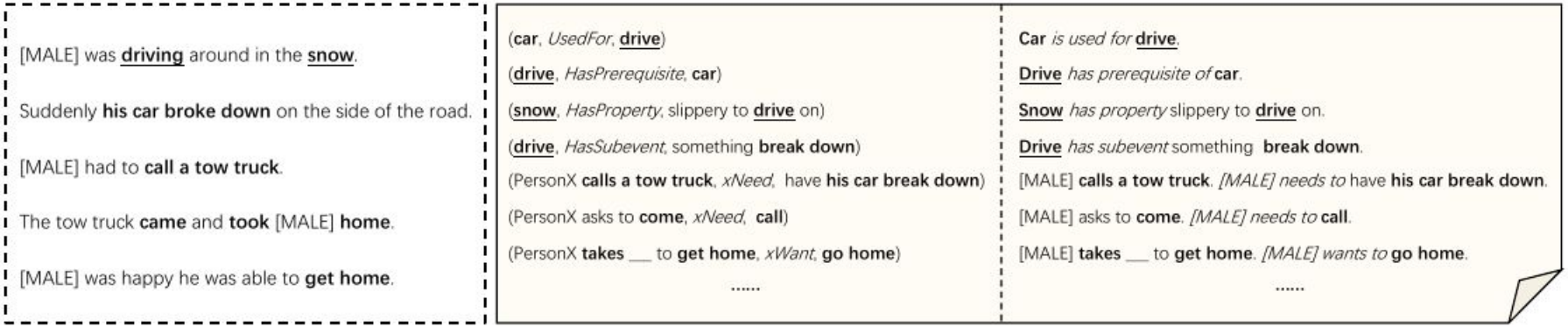


Figure 5: An example illustrating how commonsense knowledge facilitates generating reasonable stories. The right block demonstrates interrelated knowledge for the generated story, and the corresponding transformed sentences used in the training. The knowledge is retrieved from ConceptNet and ATOMIC according to the keywords denoted in **bold** in the generated story. And the underlined words represent the keywords in the leading context, while the *italic* words represent the relations.

Conclusions

- Knowledge-enhanced pretraining model with multitask learning.
- Fine-tuning on commonsense knowledge explicitly extracted from a KB leads to better performance for commonsense story generation.
- Multi-tasking learning helps the model in distinguishing true stories from fake ones.
- The results outperform strong baselines both on a quantitative and qualitative analysis.

Pre-training is (Almost) All You Need: An Application to Commonsense Reasoning

Alexandre Tamborrino, Nicola Pellicanò, Baptiste Pannier, Pascal Voitot and Louise Naudin

Samsung Strategy and Innovation Center

Introduction

Fine-tuning of pre-trained transformer models is the standard approach to many NLP tasks.

This strategy is **sub-optimal**, since the pre-trained model has no prior on the classifier labels.

In this work, a new scoring method that leverages pretrained models is applied to **commonsense reasoning**.

Commonsense Reasoning

Given a unique premise *p* and a set of hypotheses

$H = \{h_1 \dots h_n\}$, return the appropriate hypothesis h^* that matches *p*.

Usually, this kind of task is solved by classifying each pair of premise-hypothesis (p, h_i)

`[CLS] The man broke his toe. [SEP] He dropped a hammer on his foot. [SEP]` ✓

`[CLS] The man broke his toe. [SEP] He got a hole in his sock. [SEP]` ✗

The Proposed Approach

- Cast the input to a *full-text* format.

- Apply a **bidirectional word-level scoring function** that leverages the masked language model (MLM) head tuned during the pre-training phase.

Sequence Scoring Method (SSM)

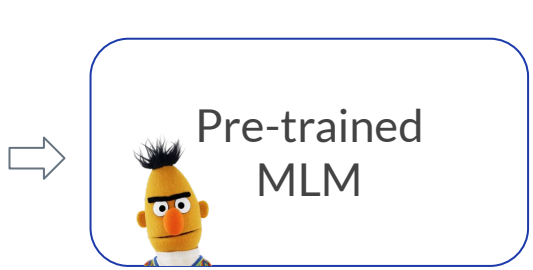
Assume we already have our *full-text* input sentence s_i , given by the transformation of premise p and hypothesis h_i into a single sentence.

Let us consider the masking of a word w in the sequence s_i . The intuition is that the **confidence** of the network in predicting w is directly **related** to the **score of** (p, h_i).

Sequence Scoring Method (SSM)

Premise *p*: The man broke his toe.
Hypothesis *h*: He dropped a hammer on his foot.

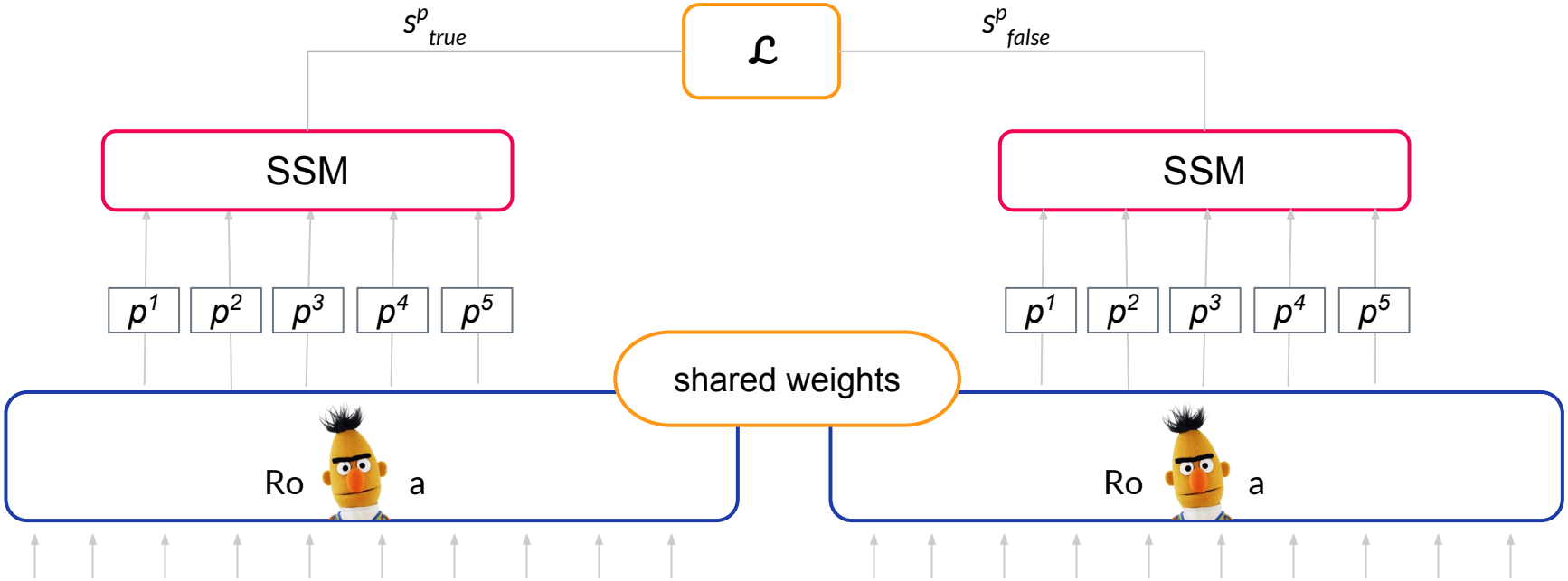
```
[CLS] [MASK] man broke his toe because he … [SEP]
[CLS] The [MASK] broke his toe because he … [SEP]
[CLS] The man [MASK] his toe because he … [SEP]
[CLS] The man broke [MASK] toe because he … [SEP]
[CLS] The man broke his [MASK] because he … [SEP]
```



$P(p^1 | s_i^{\setminus p1})$
$P(p^2 | s_i^{\setminus p2})$
$P(p^3 | s_i^{\setminus p3})$
$P(p^4 | s_i^{\setminus p4})$
$P(p^5 | s_i^{\setminus p5})$

Target premise score $s_i^p = \sum_{k=1}^{L_p} log[P(p^k | s_i^{\setminus p^k})]$

Complete model



[MASK] man broke his toe because he dropped a hammer on his foot.
The [MASK] broke his toe because he dropped a hammer on his foot.
...
The man broke his [MASK] because he dropped a hammer on his foot.

[MASK] man broke his toe because he got a hole in his sock.
The [MASK] broke his toe because he got a hole in his sock.
...
The man broke his [MASK] because he got a hole in his foot.

Complete model

N-grams sequence scoring: the SSM method can be extended to score the reconstruction of n-grams, just accumulating the scores of every gram size until n.

Loss: to better solve ranking tasks, the authors apply a margin-based loss, as already suggested in (Li et al., 2019).

Complete model

N-grams sequence scoring: the SSM method can be extended to score the reconstruction of n-grams, just accumulating the scores of every gram size until n.

Loss: to better solve ranking tasks, the authors apply a margin-based loss, as already suggested in (Li et al., 2019).

$$\mathcal{L} = \frac{1}{n} \sum_{\substack{i=1 \\ i \neq i^*}}^{n} \max\left(0, \eta - S_{i^*}^p + S_i^p\right)$$

Datasets

COPA (Gordon et al., 2012): a commonsense causal reasoning dataset where for each premise there are two candidate hypothesis.

CommonsenseQA (Talmor et al., 2019): dataset where for each commonsense question there is one correct answer and four distractors.

Swag (Zellers et al., 2018) and **HellaSwag** (Zellers et al., 2019): each premise is a video caption with four choices about what happens next in the video.

Datasets - *full-text* format

Dataset	Full-text format	Separated-sentence format
COPA (effect)	[CLS] I knocked on my neighbor's door **so** my neighbor invited me in. [SEP]	[CLS] I knocked on my neighbor's door. [SEP] My neighbor invited me in. [SEP]
COPA (cause)	[CLS] The man broke his toe **because** he dropped a hammer on his foot. [SEP]	[CLS] He dropped a hammer on his foot. [SEP] The man broke his toe. [SEP]
CommonsenseQA	[CLS] *Q:* Where on a river can you hold a cup upright to catch water on a sunny day? **A:** waterfall [SEP]	[CLS] Q: Where on a river can you hold a cup upright to catch water on a sunny day? [SEP] A: waterfall [SEP]
Swag	[CLS] We notice a man in a kayak and a yellow helmet coming in from the left. As he approaches, his kayak flips upside-down. [SEP]	[CLS] We notice a man in a kayak and a yellow helmet coming in from the left. [SEP] As he approaches, his kayak flips upside-down. [SEP]
HellaSwag	[CLS] A man is standing in front of a camera. He starts playing a harmonica for the camera. He rocks back and forth to the music as he goes. [SEP]	[CLS] A man is standing in front of a camera. He starts playing a harmonica for the camera. [SEP] He rocks back and forth to the music as he goes. [SEP]

Experiments: task probing

Remove the premises from the input. If the score on the hypothesis is better than a random baseline, the task is not solved by commonsense reasoning.

Dataset	Mode	Acc[1] (%)
COPA	hyp-only	54.6
	random	50.0
CommonsenseQA	hyp-only	22.0
	random	20.0
Swag	hyp-only	60.6
	random	25.0
HellaSwag	hyp-only	50.8
	random	25.0

Experiments: zero shot

RoBERTa$_{LARGE}$ is used in a zero shot setting on the COPA and CommonsenseQA datasets only.

Target	Grams	Accuracy
premise	1	74.0
hypothesis	1	69.8
premise	2	76.2
premise	3	79.0
premise	4	**80.0**
premise	5	79.4
BERT$_{LARGE}$	-	70.6

COPA Dataset

Target	Grams	Accuracy
premise	1	47.8
hypothesis	1	37.4
premise	2	53.2
premise	3	53.7
premise	4	**56.1**
premise	5	55.2
BERT$_{LARGE}$	-	56.7

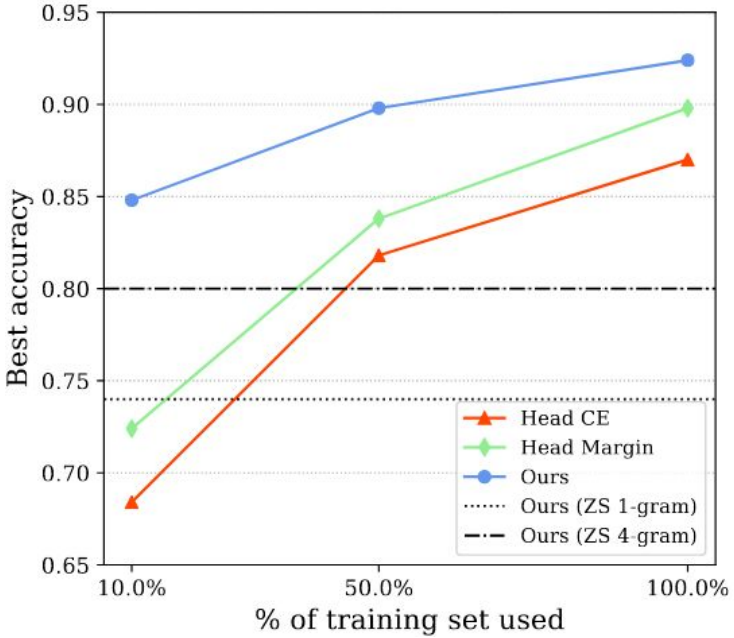
CommonsenseQA Dataset

Experiments: fine-tuning

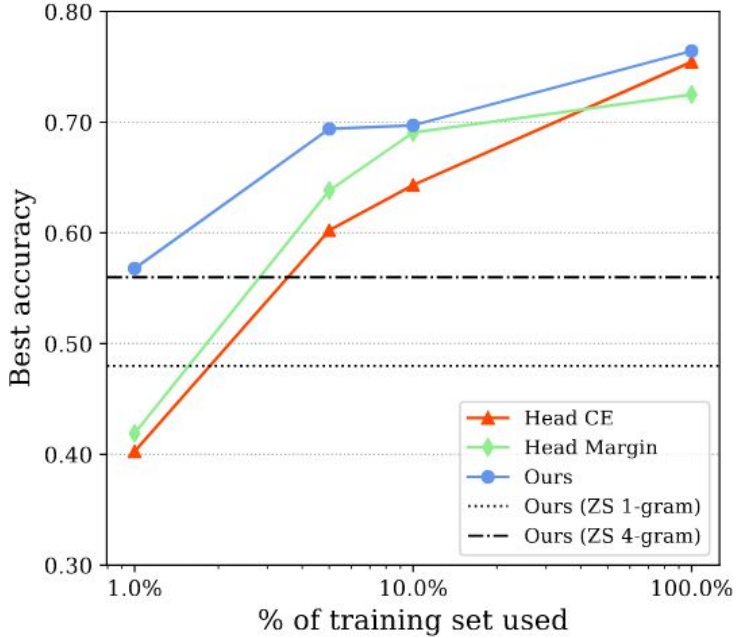
Fine-tune the scoring method while varying the percentage of training data used, comparing the results to:

- **Head CE**: Randomly initialized classifier with cross-entropy loss and *separated-sentence* format.
- **Head margin:** Randomly initialized classifier with margin-based loss and *full-text* format.

Experiments: fine-tuning results

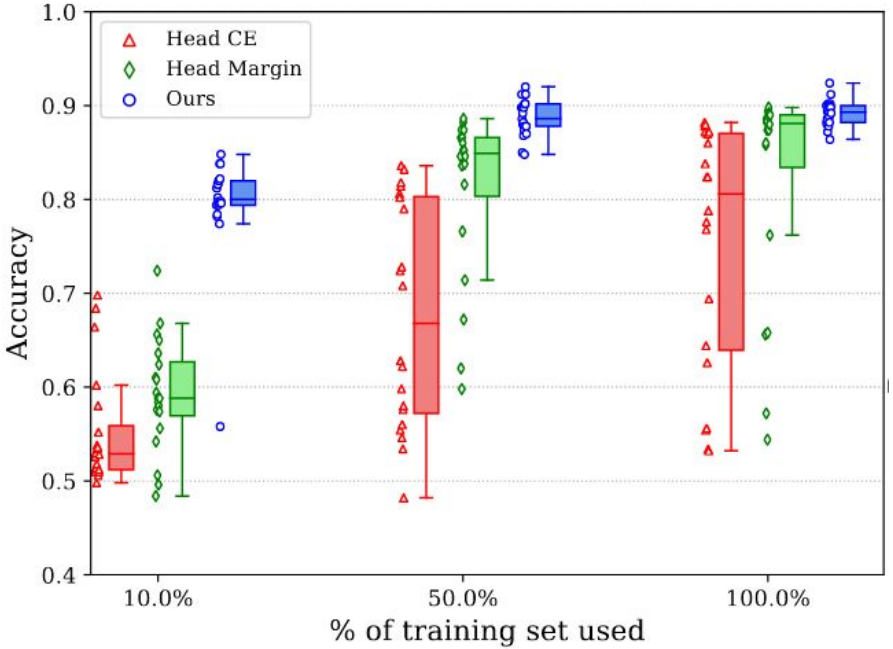


COPA

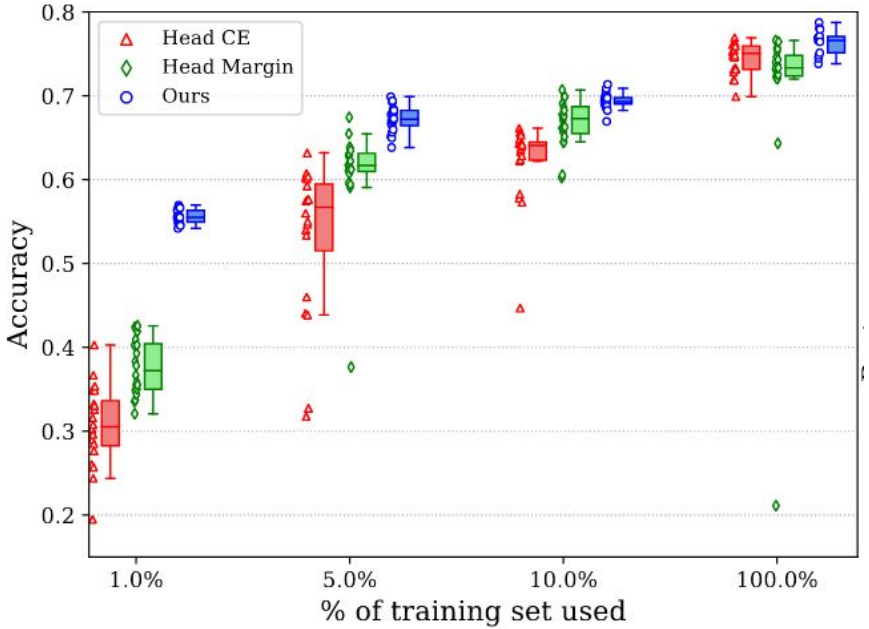


CommonsenseQA

Experiments: fine-tuning results



COPA



CommonsenseQA

Conclusions

The presented scoring function, leveraging pre-trained MLM, allows to:

- Achieve strong results on zero-shot setting, comparable to supervised approach.
- Fine-tuning with a margin-based loss the scoring function leads to good results even with few training data.
- Training is much more stable.