QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering

Michihiro Yasunaga **Hongyu Ren** **Antoine Bosselut**
Percy Liang **Jure Leskovec**
Stanford University
{myasu,hyren,antoineb,pliang,jure}@cs.stanford.edu

Motivation

- Challenges：
  - (i) Identify relevant knowledge from large KGs
  - (ii) Perform joint reasoning over the QA context and KG
- We propose a new model, QA-GNN, which addresses the above challenges through two key innovations:
  - (i) relevance scoring, where we use LMs to estimate the importance of KG nodes relative to the given QA context.
  - (ii) joint reasoning, where we connect the QA context and KG to form a joint graph, and mutually update their representations through graph-based message passing.

Task

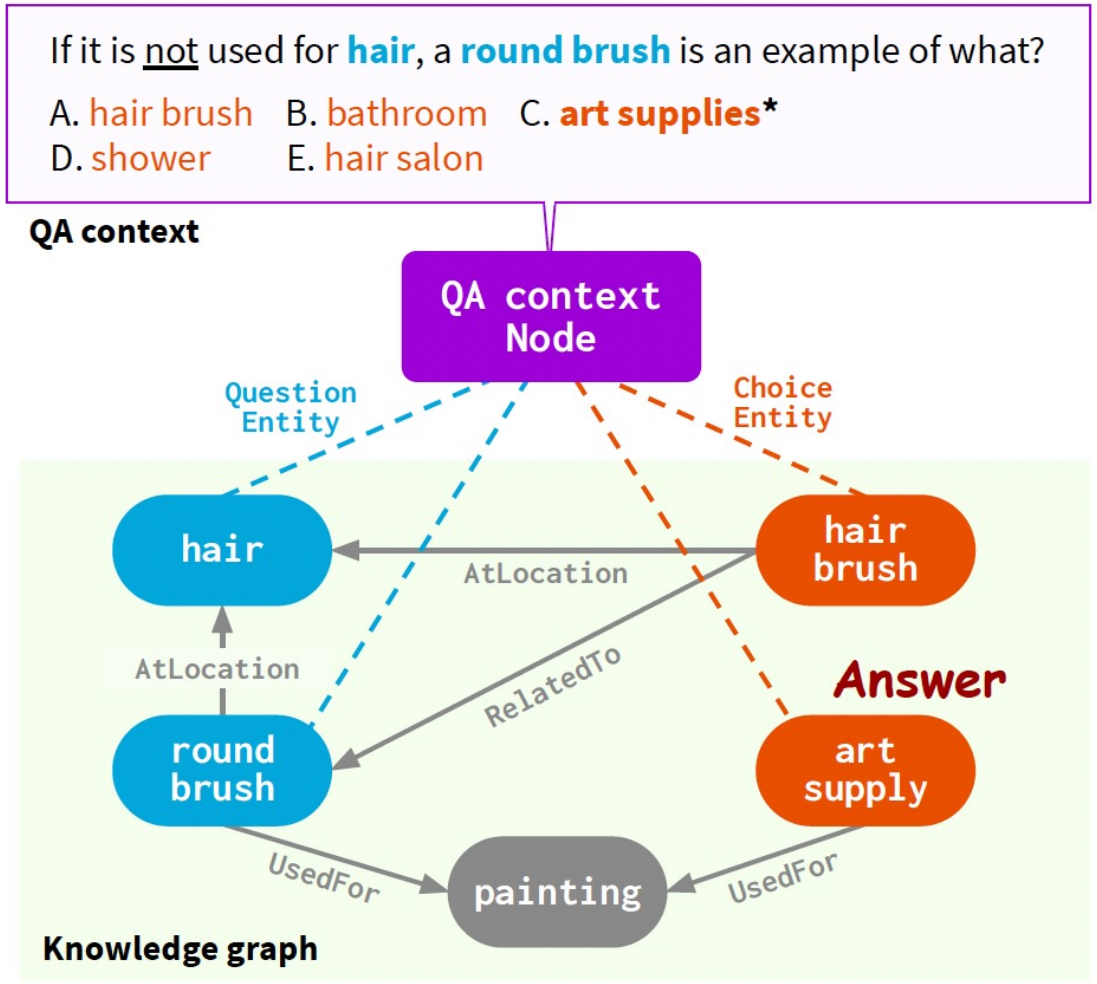


Figure 1: Given the QA context (question and answer choice; purple box), we aim to derive the answer by performing joint reasoning over the language and the knowledge graph (green box).

Framework

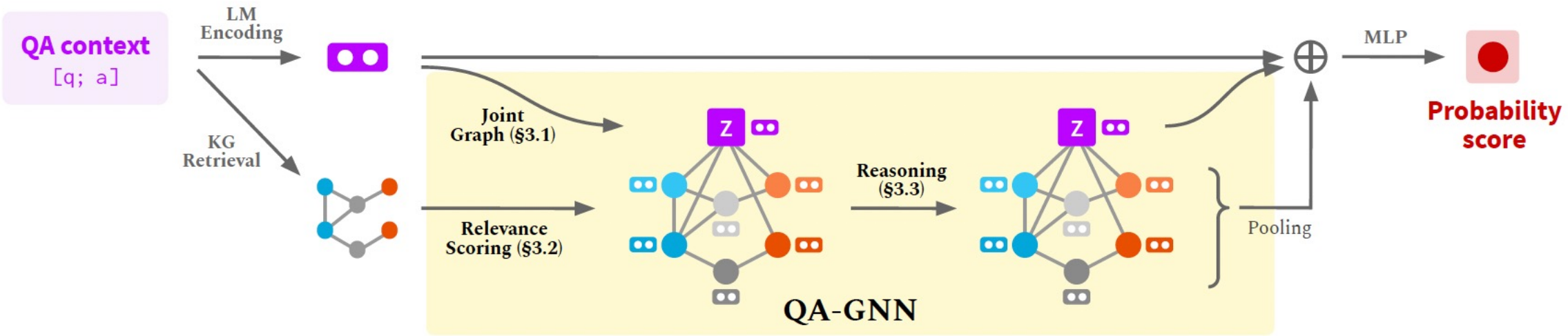


Figure 2: Overview of our approach. Given a QA context (z), we connect it with the retrieved KG to form a joint graph (*working graph*; §3.1), compute the relevance of each KG node conditioned on z (§3.2; node shading indicates the relevance score), and perform reasoning on the working graph (§3.3).

Relevance Scoring

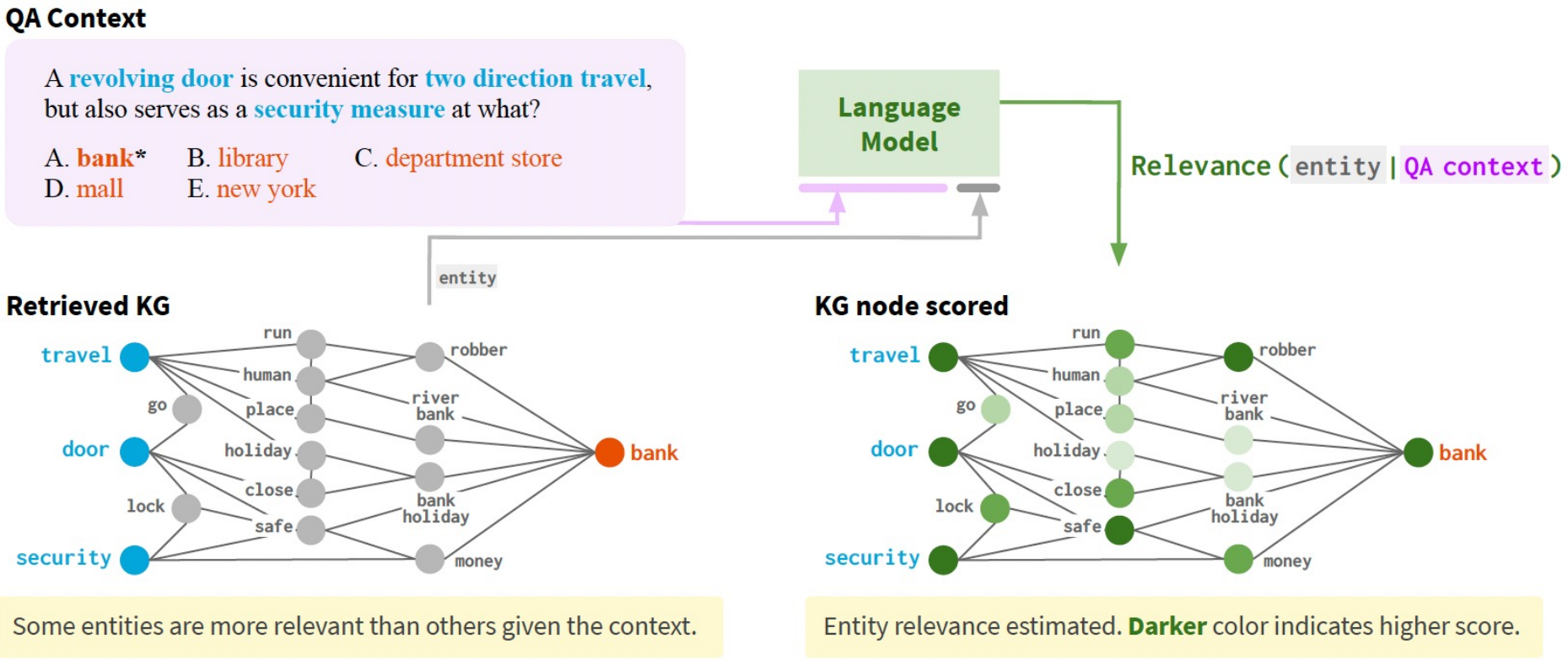


Figure 3: Relevance scoring of the retrieved KG: we use a pre-trained LM to calculate the relevance of each KG entity node conditioned on the QA context (§3.2).

Reasoning

- For each node v, we concatenate the entity *text(v)* with the QA context *text(z)* and compute the relevance score:

$$\rho_v = f_{\text{head}}(f_{\text{enc}}([\texttt{text}(z); \texttt{text}(v)]))$$

- GNN architecture:

$$\boldsymbol{h}_t^{(\ell+1)} = f_n\left(\sum_{s \in \mathcal{N}_t \cup \{t\}} \alpha_{st} \boldsymbol{m}_{st}\right) + \boldsymbol{h}_t^{(\ell)}$$

- Node type, relation, and score-aware attention:

$$\boldsymbol{\rho}_t = f_\rho(\rho_t), \qquad \begin{aligned} \boldsymbol{q}_s &= f_q(\boldsymbol{h}_s^{(\ell)}, \boldsymbol{u}_s, \boldsymbol{\rho}_s), \\ \boldsymbol{k}_t &= f_k(\boldsymbol{h}_t^{(\ell)}, \boldsymbol{u}_t, \boldsymbol{\rho}_t, \boldsymbol{r}_{st}), \end{aligned} \qquad \alpha_{st} = \frac{\exp(\gamma_{st})}{\sum_{t' \in \mathcal{N}_s \cup \{s\}} \exp(\gamma_{st'})}, \quad \gamma_{st} = \frac{\boldsymbol{q}_s^\top \boldsymbol{k}_t}{\sqrt{D}}$$

Dataset

- CommonsenseQA
  - a 5-way multiple choice QA task that requires reasoning with commonsense knowledge, containing 12,102 questions. The test set of CommonsenseQA is not publicly available

- OpenBookQA
  - a 4-way multiple choice QA task that requires reasoning with elementary science knowledge, containing 5,957 questions.

Result

Methods	IHdev-Acc. (%)	IHtest-Acc. (%)
RoBERTa-large (w/o KG)	73.07 (±0.45)	68.69 (±0.56)
+ RGCN (Schlichtkrull et al., 2018)	72.69 (±0.19)	68.41 (±0.66)
+ GconAttn (Wang et al., 2019a)	72.61(±0.39)	68.59 (±0.96)
+ KagNet (Lin et al., 2019)	73.47 (±0.22)	69.01 (±0.76)
+ RN (Santoro et al., 2017)	74.57 (±0.91)	69.08 (±0.21)
+ MHGRN (Feng et al., 2020)	74.45 (±0.10)	71.11 (±0.81)
+ QA-GNN (**Ours**)	**76.54** (±0.21)	**73.41** (±0.92)

Table 2: **Performance comparison on *Commonsense QA* in-house split** (controlled experiments). As the official test is hidden, here we report the in-house Dev (IHdev) and Test (IHtest) accuracy, following the data split of Lin et al. (2019).

Methods	Test
RoBERTa (Liu et al., 2019)	72.1
RoBERTa+FreeLB (Zhu et al., 2020) (ensemble)	73.1
RoBERTa+HyKAS (Ma et al., 2019)	73.2
RoBERTa+KE (ensemble)	73.3
RoBERTa+KEDGN (ensemble)	74.4
XLNet+GraphReason (Lv et al., 2020)	75.3
RoBERTa+MHGRN (Feng et al., 2020)	75.4
Albert+PG (Wang et al., 2020b)	75.6
Albert (Lan et al., 2020) (ensemble)	76.5
UnifiedQA* (Khashabi et al., 2020)	**79.1**
RoBERTa + QA-GNN (**Ours**)	76.1

Table 3: **Test accuracy on *CommonsenseQA*'s official leaderboard**. The top system, UnifiedQA (11B parameters) is 30x larger than our model.

Result

Methods	RoBERTa-large	AristoRoBERTa
Fine-tuned LMs (w/o KG)	64.80 (±2.37)	78.40 (±1.64)
+ RGCN	62.45 (±1.57)	74.60 (±2.53)
+ GconAtten	64.75 (±1.48)	71.80 (±1.21)
+ RN	65.20 (±1.18)	75.35 (±1.39)
+ MHGRN	66.85 (±1.19)	80.6
+ QA-GNN (**Ours**)	**70.58** (±1.42)	**82.77** (±1.56)

Table 4: **Test accuracy comparison on *OpenBook QA*** (controlled experiments). Methods with AristoRoBERTa use the textual evidence by Clark et al. (2019) as an additional input to the QA context.

Methods	Test
Careful Selection (Banerjee et al., 2019)	72.0
AristoRoBERTa	77.8
KF + SIR (Banerjee and Baral, 2020)	80.0
AristoRoBERTa + PG (Wang et al., 2020b)	80.2
AristoRoBERTa + MHGRN (Feng et al., 2020)	80.6
Albert + KB	81.0
T5* (Raffel et al., 2020)	83.2
UnifiedQA* (Khashabi et al., 2020)	**87.2**
AristoRoBERTa + QA-GNN (**Ours**)	82.8

Table 5: **Test accuracy on *OpenBookQA* leaderboard**. All listed methods use the provided science facts as an additional input to the language context. The top 2 systems, UnifiedQA (11B params) and T5 (3B params) are 30x and 8x larger than our model.

Ablation

Graph Connection (§3.1)	Dev Acc.
No edge between Z and KG nodes	74.81
Connect Z to all KG nodes	76.38
Connect Z to QA entity nodes (**final**)	**76.54**

Relevance scoring (§3.2)	Dev Acc.
Nothing	75.56
w/ contextual embedding	76.31
w/ relevance score (**final**)	**76.54**
w/ both	76.52

GNN Attention & Message (§3.3)	Dev Acc.
Node type, relation, score-aware (**final**)	**76.54**
- type-aware	75.41
- relation-aware	75.61
- score-aware	75.56

GNN Layers (§3.3)	Dev Acc.
$L = 3$	75.53
$L = 4$	76.34
$L = 5$ (**final**)	**76.54**
$L = 6$	76.21
$L = 7$	75.96

Table 6: **Ablation study** of our model components, using the CommonsenseQA IHdev set.

Case

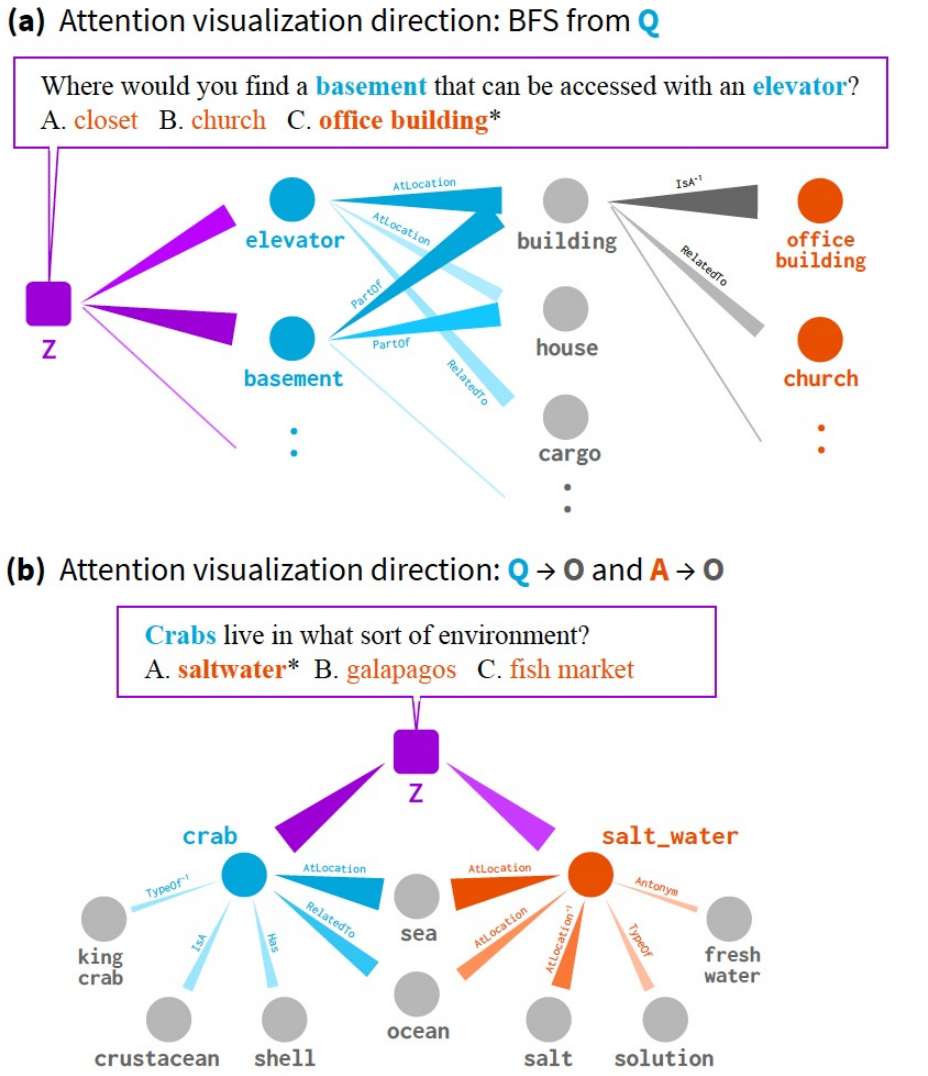


Figure 4: **Interpreting QA-GNN's reasoning process** by analyzing the node-to-node attention weights induced by the GNN. Darker and thicker edges indicate higher attention weights.

Structured reasoning

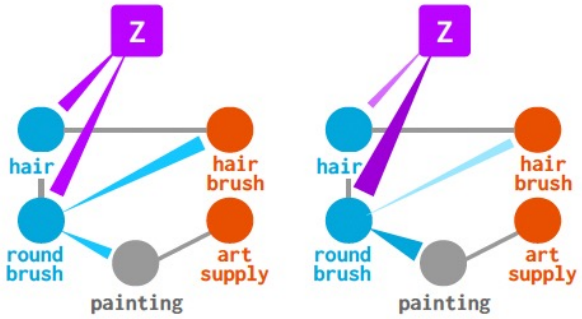
Methods	**IHtest-Acc.** (Overall)	**IHtest-Acc.** (Question w/ **negation**)
RoBERTa-large (w/o KG)	68.7	54.2
+ KagNet	69.0 (+0.3)	54.2 (+0.0)
+ MHGRN	71.1 (+2.4)	54.8 (+0.6)
+ QA-GNN (**Ours**)	73.4 (+4.7)	**58.8 (+4.6)**
+ QA-GNN (no edge between Z and KG)	71.5 (+2.8)	55.1 (+0.9)

Table 7: Performance on **questions with negation** in *CommonsenseQA*. () shows the difference with RoBERTa. Existing LM+KG methods (KagNet, MHGRN) provide limited improvements over RoBERTa (+0.6%); QA-GNN exhibits a bigger boost (+4.6%), suggesting its strength in structured reasoning.

Structured reasoning

Original Question

If it is **not** used for **hair**, a **round brush** is an example of what?
A. hair brush B. **art supply***



GNN **1st** Layer

GNN **Final** Layer

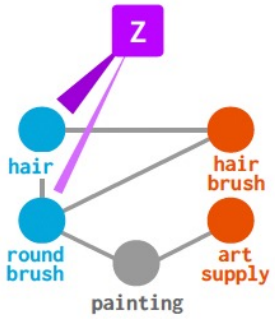
A. hair brush (0.38)

B. art supply (0.64)

Model Prediction

(a) Negation Flipped

If it is used for **hair**, a **round brush** is an example of what? A. hair brush B. art supply



GNN **Final** Layer

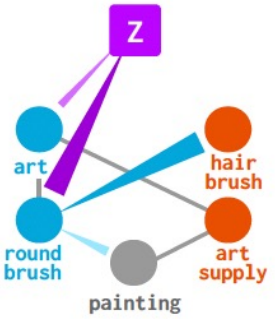
A. hair brush (0.81)

B. art supply (0.19)

Model Prediction

(b) Entity Changed (hair → art)

If it is **not** used for **art**, a **round brush** is an example of what? A. hair brush B. art supply



GNN **Final** Layer

A. hair brush (0.72)

B. art supply (0.28)

Model Prediction

Structured reasoning

Example (Original taken from *CommonsenseQA* Dev)	**RoBERTa Prediction**	**Our Prediction**
[Original] If it is **not** used for hair, a round brush is an example of what? A. hair brush B. art supply	A. hair brush (✗)	B. art supply (✓)
[Negation flip] If it is used for hair, a round brush is an example of what?	A. hair brush (✓ just no change?)	A. hair brush (✓)
[Entity change] If it is not used for art a round brush is an example of what?	A. hair brush (✓ just no change?)	A. hair brush (✓)
[Original] If you have to read a book that is very dry you may become what? A. interested B. bored	B. bored (✓)	B. bored (✓)
[Negation ver 1] If you have to read a book that is very dry you may **not** become what?	B. bored (✗)	A. interested (✓)
[Negation ver 2] If you have to read a book that is **not** dry you may become what?	B. bored (✗)	A. interested (✓)
[Double negation] If you have to read a book that is **not** dry you may **not** become what?	B. bored (✓ just no change?)	A. interested (✗)

Semantic Frame Forecast

Chieh-Yang Huang and Ting-Hao (Kenneth) Huang
Pennsylvania State University, University Park, PA 16802, USA
{chiehyang, txh710}@psu.edu

Task

- This paper introduces semantic frame forecast, a task that predicts the semantic frames that will occur in the next 10, 100, or even 1,000 sentences in a running story.

- Prior work focused on predicting the immediate future of a story, such as one to a few sentences ahead. However, when novelists write long stories, generating a few sentences is not enough to help them gain high-level insight to develop the follow-up story.

Task

- In this paper, we formulate a long story as a sequence of "**story blocks**," where each block contains a fixed number of sentences (e.g., 10, 100, or 200).

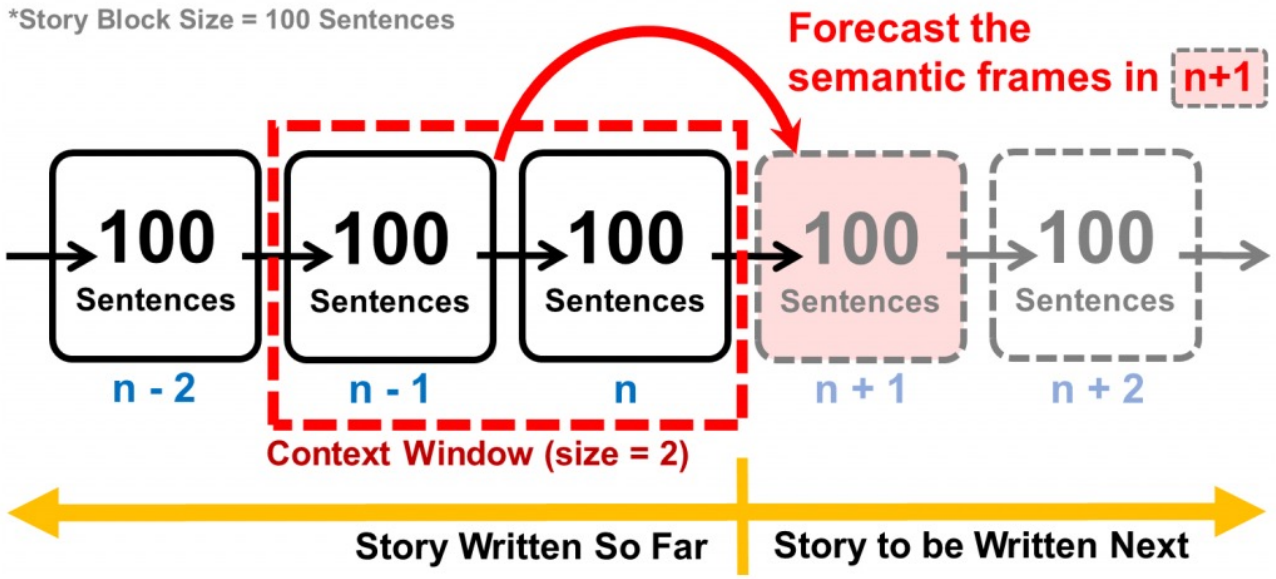
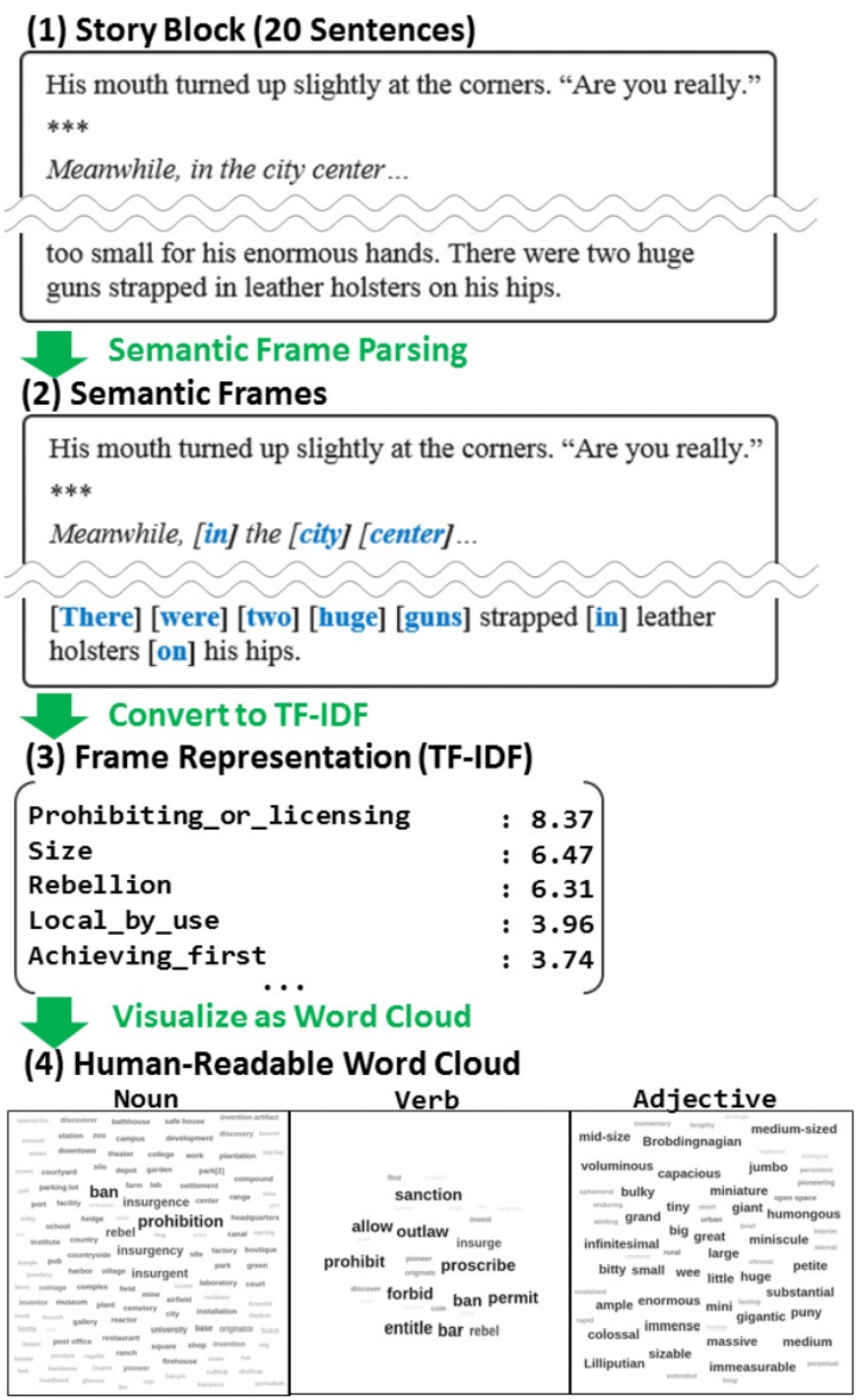


Figure 1: The semantic frame forecast is a task that predicts the semantic frames that will occur in the next part of a story based on the texts written so far.

Task

- Evaluation Metric
  - Cosine Similarity
  - Human evaluation



Data

- Bookcorpus Dataset
  - 15, 605 raw books and their corresponding meta data.

    - (i) short books whose size is less than 10KB;
    - (ii) books that contain HTML code;
    - (iii) books that are in the epub format (an e-book file format);
    - (iv) books that are not in English; (v) books that are in the "Non-Fiction" genre;
    - (vi) books that are in the "Anthologies" genre;
    - (vii) books that are in the "Graphic Novels & Comics" genre.

  - a total of 4, 794 books

Data

Block Size	**5**	**10**	**20**	**50**	**100**	**150**	**200**	**300**	**500**	**1000**
# Words Mean	71.7	143.5	286.9	717.2	1433.9	2149.8	2865.3	4293.7	7142.5	14212.3
# Frames Mean	17.5	35.0	69.9	174.5	348.6	522.1	695.4	1040.7	1727.3	3417.2
# Events Mean	10.0	20.0	39.9	99.8	199.4	298.9	398.2	596.4	991.2	1967.1
# Train	3,744,948	1,869,947	932,464	369,941	182,479	119,967	88,720	57,455	32,469	13,749
# Valid	574,840	287,054	143,166	56,838	28,073	18,466	13,672	8,881	5,035	2,166
# Test	1,054,816	526,687	262,625	104,198	51,396	33,776	24,987	16,178	9,138	3,861

Table 1: The statistic of Bookcorpus dataset in ten different story block lengths. We use Open-Sesame to parse the semantic frame for each sentence. The *Events* represents the SVO tuples (Martin et al., 2018).

Data

- CODA-19 Dataset.
  - contains 10, 966 human-annotated English abstracts for five different aspects: **Background**, **Purpose**, **Method**, **Finding/Contribution**, and **Other**.

    - (i) aspects：other
    - (ii) Abstracts that contain Unicode characters

  - a total of 7, 962 Abstract

Data

Block Size	1	3	5
# Words Mean	26.3	77.3	124.7
# Frames Mean	6.0	17.5	27.6
# Events Mean	1.2	3.5	5.6
# Train	48,489	9,858	2,739
# Valid	5,615	1,146	334
# Test	5,238	1,047	287

Table 2: The statistic of CODA-19 dataset in three different story block lengths. We use Open-Sesame to parse the semantic frame for each sentence. The *Events* represents the SVO tuples (Martin et al., 2018).

Model

- Baseline

Replay Model. For each instance, the replay model takes the frame representation in the n-th story block as the prediction, *i.e.*, the same frames will occur again.

Prior Model. The prior model computes the mean of the frame representation over the training set and uses it as the prediction for all the testing instances.

- Best model

BERT. We take the pure text in the foregoing story block as the feature and apply the pretrained BERT model (Devlin et al., 2019). BERT has a token length limitation, so we set the maximum length of tokens to 500 for Bookcorpus and 300 for CODA-19. Sentences with more than 500 tokens are truncated from the left. We take the `[CLS]` token representation from the last layer and add a dense layer on top of it to predict the follow-up frame representation. The model is trained with a learning rate of 1e-5 and a batch size of 32. We optimize the model using the cosine distance and apply the early stopping when no improvement for five epochs. The model with the best score on the validation set is kept for testing.

Result

Feature	Model	Block Size									
		5	**10**	**20**	**50**	**100**	**150**	**200**	**300**	**500**	**1000**
-	Replay Baseline	.0654	.0915	.1237	.1737	.2163	.2448	.2665	.3000	.3462	.4155
-	Prior Baseline	.2029	.2435	.2857	.3389	.3754	.3962	.4105	.4302	.4528	.4776
Frame	IR Baseline	-	.0631	.0851	.1290	.1841	.2085	.2262	.2536	.2859	.3321
Frame	Random Forest	.2037	.2448	.2881	.3427	.3807	.4025	.4184	.4402	.4659	.4966
Frame	LGBM	.2072	.2506	.2967	.3564	.3995	**.4255**	**.4441**	**.4711**	**.5048**	**.5510**
Frame	DAE	.2082	.2515	.2966	.3547	.3976	.4223	.4400	.4598	.4898	.5280
Event	Event-Rep	.2111	.2541	.2994	.3532	.3929	.4126	.4280	.4453	.4626	.4792
Text	BERT	**.2172**	**.2611**	**.3073**	**.3637**	**.4012**	.4229	.4371	.4559	.4779	.5057
Text	GPT-2	.0519	.0739	.0990	.1402	-	-	-	-	-	-
	DELTA	.0142	.0176	.0216	.0249	.0257	.0293	.0336	.0409	.0520	.0734

Table 3: Baseline result for Bookcorpus dataset. BERT and Event-Rep work better in smaller block sizes, while models using frame representation perform better in larger block sizes. DELTA represents the difference between the best model and the prior baseline — an extremely simple but strong baseline — in that specific block size. The small value of DELTA shows that semantic frame forecast is challenging yet possible.

Result

Feature	Model	Block Size 1	3	5
-	Replay Baseline	.0524	.0971	.1363
-	Prior Baseline	.1573	.2067	.2288
Frame	IR Baseline	.0315	.0601	.0752
Frame	Random Forest	.1581	.2081	.2278
Frame	LGBM	.1561	.2024	.2094
Frame	DAE	.1611	.2155	**.2380**
Event	Event-Rep	.1595	.2118	.2332
Text	BERT	.1660	.2202	.2353
Text	SciBERT	**.1675**	**.2219**	.2339
	DELTA	.0102	.0152	.0092

Table 4: Baseline result for CODA-19 dataset. SciBERT performs the best in block size 1 and 3. Using the frame representation as the feature, DAE performs the best for block size 5. DELTA shows the difference between the best model and the prior baseline in that specific block size. The small value of DELTA shows that semantic frame forecast is challenging yet possible.

Result

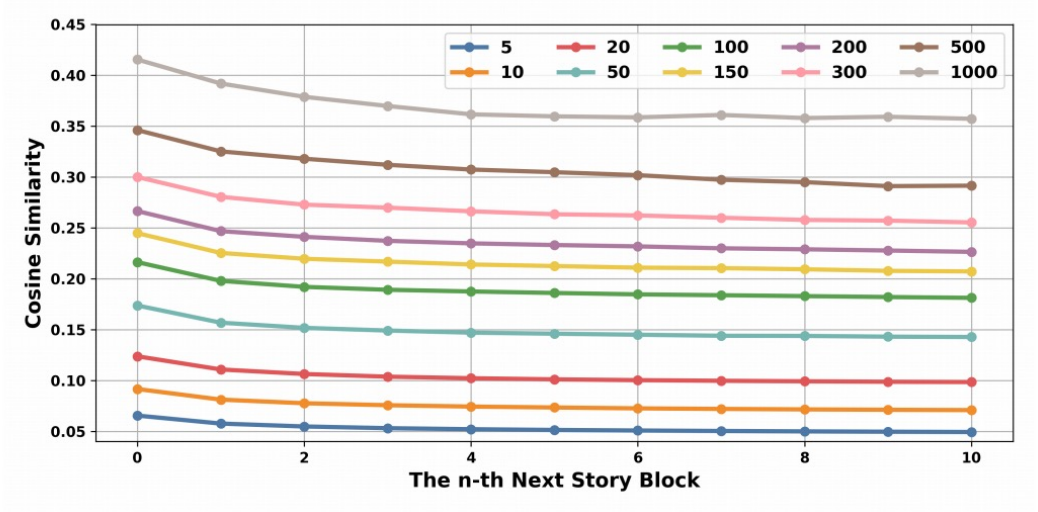


Figure 3: Using the replay baseline to predict the n+i-th story block from the n-th story block (story block size = 5, 10, $\cdots$, 1000.) Things that happen in the current story block are more likely to happen again shortly.

Feature	Model	Block Size 5	10	20	50	100	150	200
-	Prior	.2029	.2435	.2856	.3388	.3754	.3962	.4105
Frame	IR	.0401	.0615	.0900	.1368	.1775	.2051	.2262
Frame	RF	.2030	.2440	.2871	.3418	.3801	.4025	.4184
Frame	LGBM	.2033	.2472	.2935	.3540	**.3980**	**.4248**	**.4441**
Frame	DAE	.2058	.2482	.2929	.3507	.3926	.4178	.4400
Event	Event-Rep	.2046	.2470	.2905	.3454	.3799	.4069	.4171
Text	BERT	**.2088**	**.2529**	**.2981**	**.3550**	.3949	.4178	.4371

Table 5: Result of the downsampling experiment. Although all the performance drops, the observations we find are still true. Therefore, the conclusions are not merely caused by the effect of data size.

Analyse

- “Prior” is a robust and strong baseline
- Replay baseline shows the relation of consecutive story blocks.
- Event-Rep works better in short stories.
- BERT performs very well in short stories.
- The good performance does not merely come from the number of instances.