



北京航空航天大學
BEIHANG UNIVERSITY

自然语言处理

人工智能研究院

主讲教师 沙磊



在语言模型
中引入知识

Contents

- Techniques to add knowledge to LMs
 - 1. Add pretrained entity embeddings
 - 2. Use an external memory
 - 3. Modify the training data
- Evaluating knowledge in LMs

回顾：语言模型

- 标准语言模型，给出文本序列，预测下一个词
 - The students opened their books.
- 掩码语言模型（如BERT），根据双向的文本信息预测被掩码的 token
 - I [MASK] to the [MASK].

went

store

- 如上两种语言模型，都可以利用大量无标注语料库训练

回顾：语言模型用途

- 语言模型的用途：生成、评估句子的概率
 - Summarization
 - Dialogue
 - Autocompletion
 - Machine translation
 - Fluency evaluation
- 如今，LM也可以用来生成一些预训练的文本表示用来编码文本中的语义信息，提供给下游NLP任务使用
 - Text classification
 - Question answering
- 如果LM用海量文本预训练，可不可以直接被用作知识库？

What does a language model already know?

- iPod Touch is produced by _____.
- London Jazz Festival is located in _____.
- Dani Alves plays with _____.
- Carl III used to communicate in _____.
- Ravens can _____.

Check out what BERT-Large predicts

What does a language model already know?

- iPod Touch is produced by Apple.
- London Jazz Festival is located in London .
预测看似合理，不一定符合事实
- Dani Alves plays with Santos ->Barcelona
- Carl III used to communicate in German ->Swedish
- Ravens can fly .

Check out what BERT-Large predicts

What does a language model know?

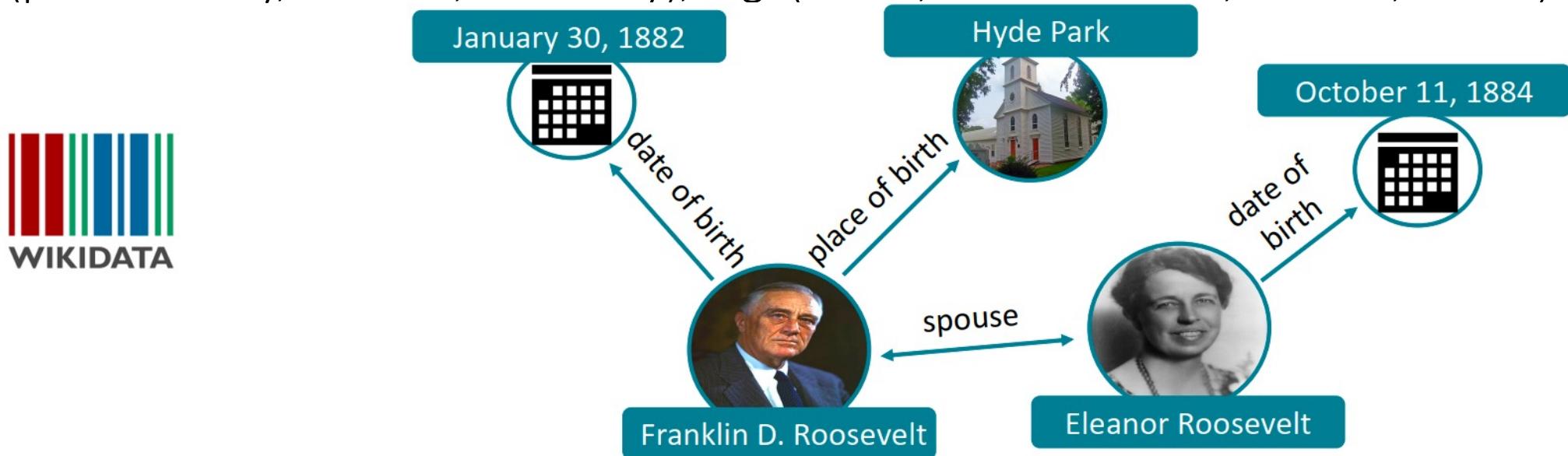
- Observation: predictions generally make sense (e.g. the correct types), but are **not all factually correct**.
- Why might this happen?
 - **Unseen facts**: some facts may not have occurred in the training corpora at all
 - **Rare facts**: LM hasn't seen enough examples during training to memorize the fact
 - **Model sensitivity**: LM may have seen the fact during training, but it was phrased in a different way than how we are testing, so the LM is confused
 - Fails to answer "x was created in y" but correctly answers "x was made in y"
- Takeaway: LMs have some knowledge, but **fail to reliably recall knowledge**
 - We will talk about how to address this key challenge facing LMs!

Why do we want to build knowledge-aware language models?

- LM's pretrained representations can benefit downstream tasks that leverage knowledge
 - e.g. Question answering and relation extraction (extracting the relations between two entities in a sentence) are much easier with knowledge about the entities
 - We'll come back to this when we talk about evaluation!
- Stretch goal: can LMs ultimately replace traditional knowledge bases?
 - Instead of querying a knowledge base with formal query (e.g. SQL), query the LM with a natural language prompt!
 - Of course, this requires LM to have high quality on recalling facts, and this is an active area of research

Traditional knowledge bases and how to query them

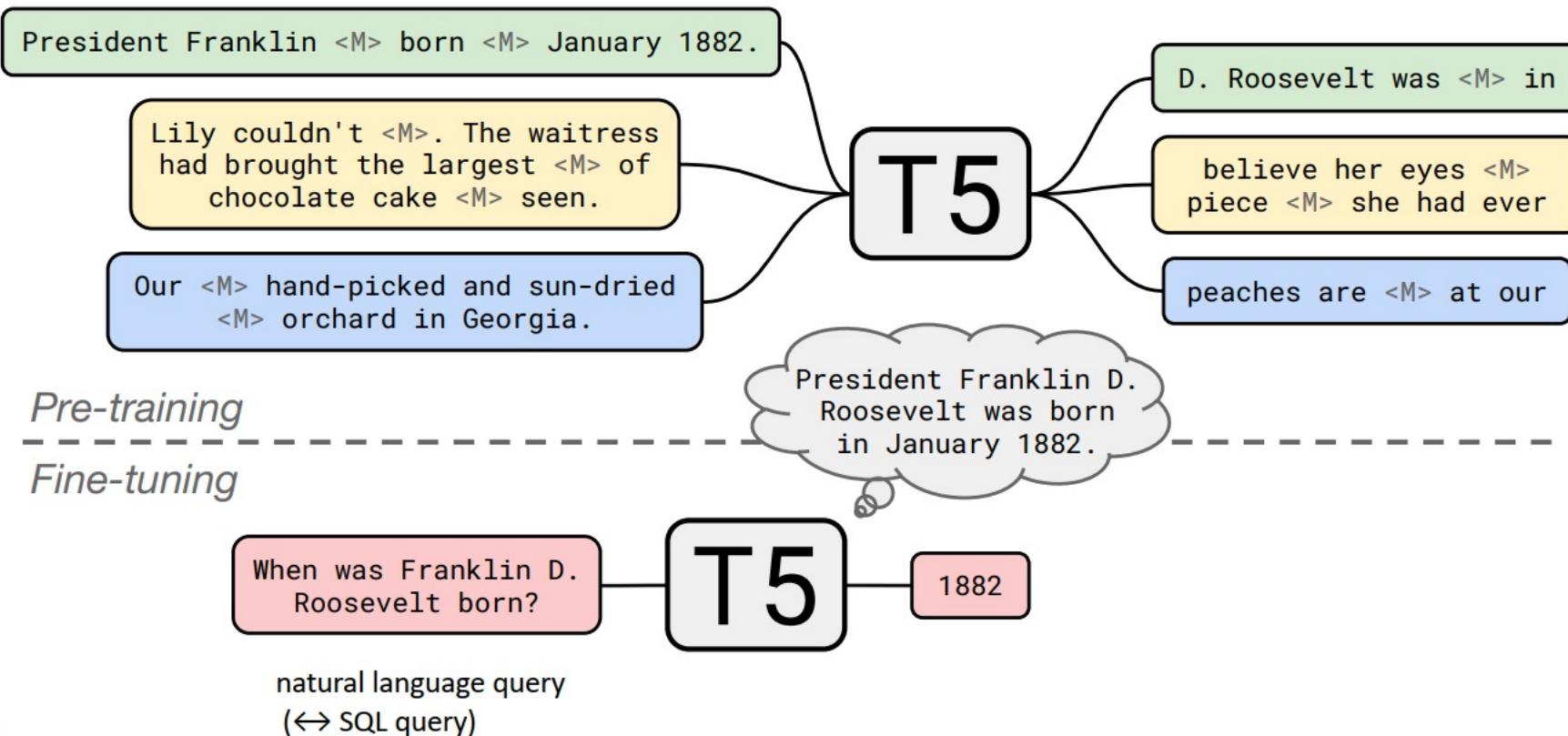
- Each Knowledge base entry can be written as a triple:
 - (parent entity, relation, tail entity), e.g. (“FDR”, “date of birth”, “Jan 30, 1882”)



- You can query knowledge base with a formal query such as SQL statement:
 - “What is the date of birth of FDR?”

How to query language models as knowledge bases

- Pretrain LM over unstructured text and then query with natural language.



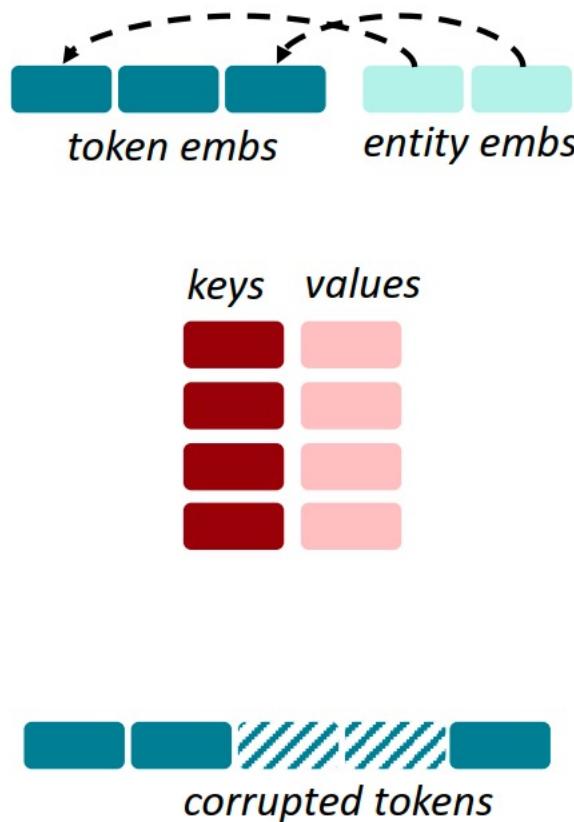
Advantages of using language models over traditional KBs

- LMs can be pretrained over large amounts of **unstructured and unlabeled text**
 - KBs typically require manual annotation
- LMs support more **flexible natural language queries**
 - Example: What does the final F in the song U.F.O.F. stand for?
 - Traditional KB may not have a specific relation “final F”; LM may learn it implicitly
- However, there are also many open challenges to using LMs as KBs:
 - **Hard to interpret** (it’s unclear why LM produces this answer KB has provenance)
 - Hard to trust (LM may produce a realistic but incorrect answer KB either returns the correct answer or returns no answer)
 - Hard to modify (hard to update knowledge in LM KB is directly editable)
 - => Open up exciting opportunities for further research!



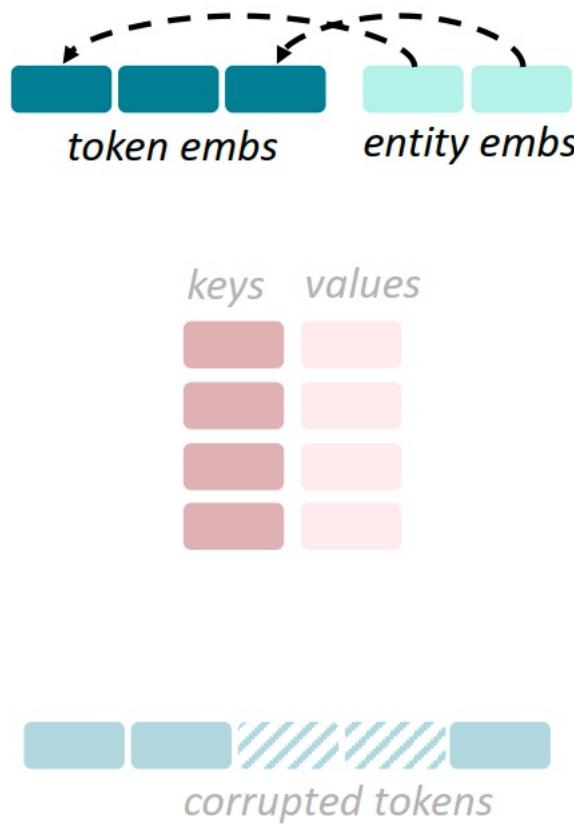
如何将知识加入LM中？

Techniques to add knowledge to LMs



- Add pretrained entity embeddings
 - ERNIE
 - QAGNN/GreaseLM
- Use an external memory
 - KGLM
- Modify the training data
 - WKLM
 - ERNIE (another!), salient span masking

Techniques to add knowledge to LMs



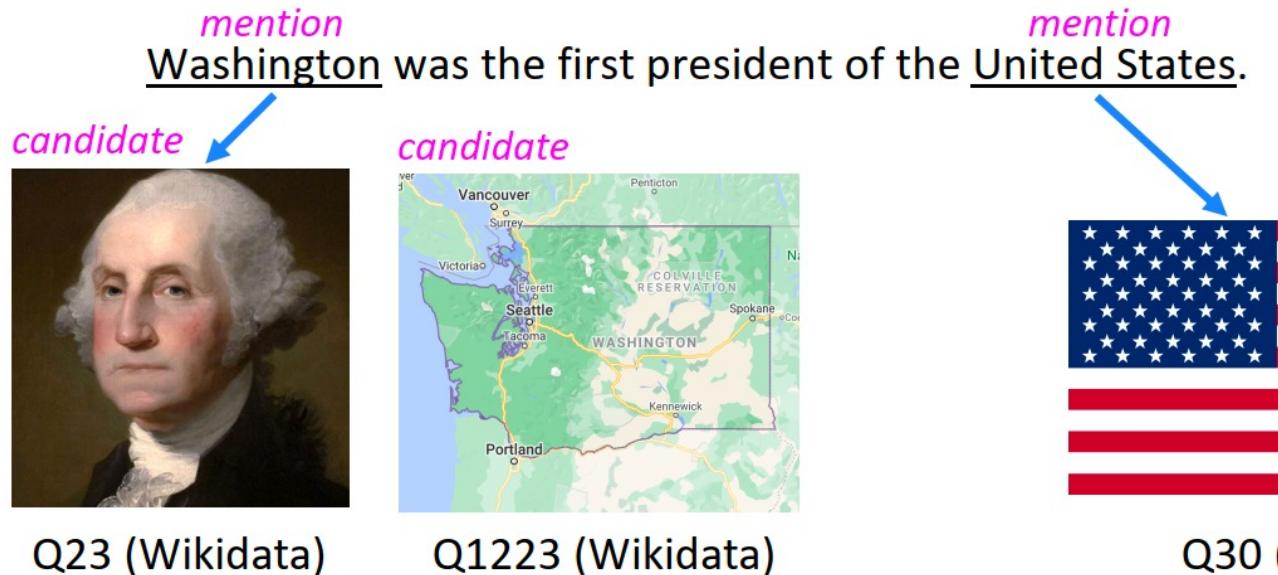
- Add pretrained entity embeddings
 - ERNIE
 - QAGNN/GreaseLM
- Use an external memory
 - KGLM
- Modify the training data
 - WKLM
 - ERNIE (another!), salient span masking

Method 1: Add pretrained (entity) embeddings

- Observation: Facts about the world are usually in terms of **entities**
 - Example: Washington was the first president of the United States.
- However, the typical word embeddings we use do **not** have a notion of entities
 - We use **different word embeddings** for “U.S.A.”, “United States of America” and “America” even though they all refer to the same entity
- What if we assign a single embedding per entity?
 - **Single entity embedding** for “U.S.A.”, “United States of America” and “America”
- **Goal:** Get pretrained entity embeddings that encode factual knowledge, and add to language model
- Note: To use entity embeddings for text, we need to do a task **called entity linking**

Aside: What is entity linking?

- Link **mentions** in text to **entities** in a knowledge base



(It's like looking up word in word embedding dictionary. But looking up entity mention in KB is a bit trickier, because string matching may not work..)



- Entity linking involves resolving ambiguous mentions (e.g. using context)
- Entity linking tells us which entity embeddings are relevant to the text

Method 1: Add pretrained entity embeddings

- Summary: Entity embeddings are like word embeddings, but for entities in a knowledge base!

$$\text{George Washington} = \begin{pmatrix} 0.111 \\ -0.345 \\ 0.876 \\ -0.201 \end{pmatrix}$$

- Many techniques for training entity embeddings:
 - Knowledge graph embedding methods (e.g., [TransE](#))
 - Word-entity co-occurrence methods (e.g., [Wikipedia2Vec](#))
 - Transformer encodings of entity descriptions (e.g., [BLINK](#))
- Any of those entity embeddings can be used for the knowledge integration methods we will talk about today

Method 1: Add pretrained entity embeddings

- Question: How do we incorporate pretrained entity embeddings when they're from an **different embedding space** than the language model?
- Answer: Learn a **fusion layer h** that combines word info (from LM) and entity info

$$\mathbf{h}_j = F(\mathbf{W}_t \mathbf{w}_j + \mathbf{W}_e \mathbf{e}_k + b)$$

- \mathbf{w}_j is the embedding of word - in a sequence of words
- \mathbf{e}_k is the corresponding entity embedding
- Intuition: there's alignment between entities and words in the sentence such that projections $\mathbf{W}_t \mathbf{w}_j$ and $\mathbf{W}_e \mathbf{e}_k$ are in the same vector space

ERNIE: Enhanced Language Representation with Informative Entities[Zhang et al., ACL 2019]

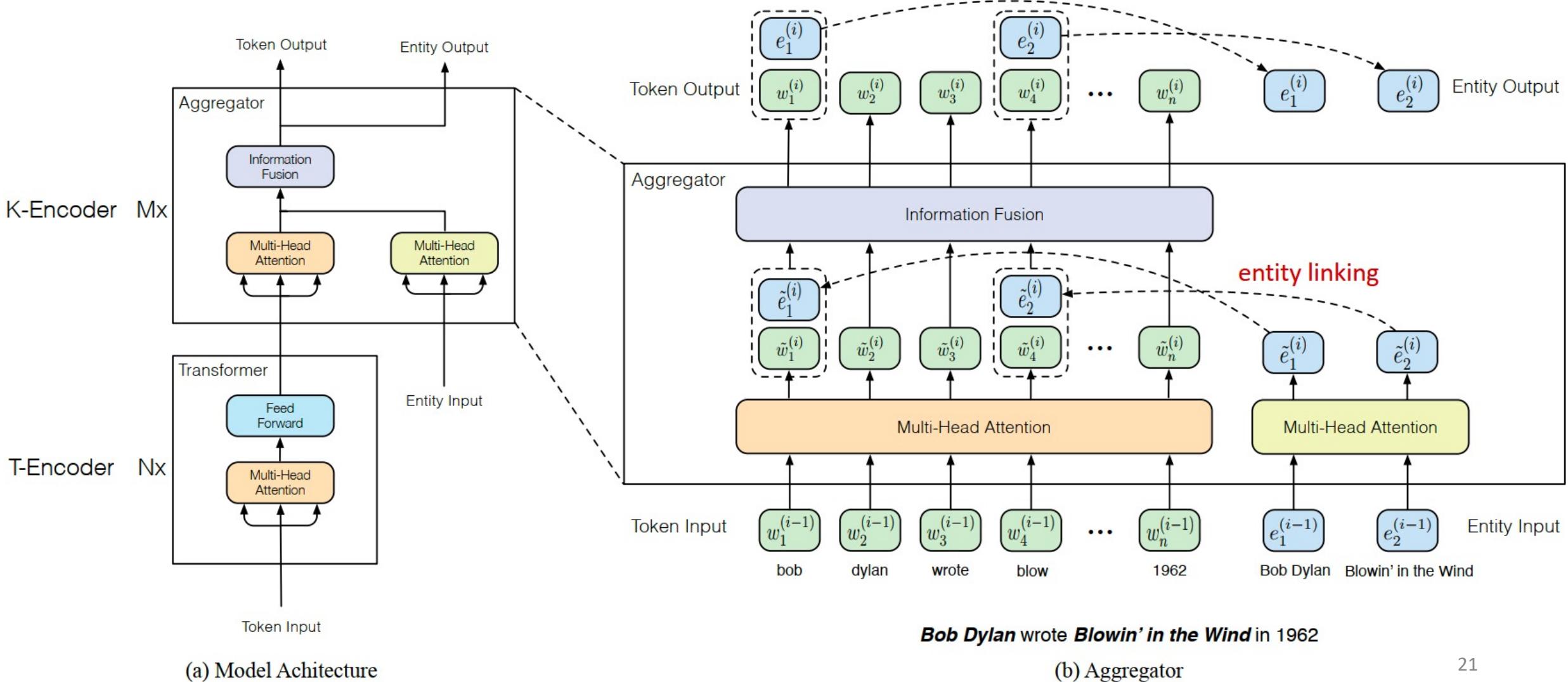
- **Text encoder**: multi-layer bidirectional Transformer encoder over the token in the sentence
- **Knowledge encoder**: each block is composed of:
 - Two **self-attention layers** – one for entity embeddings and one for token embeddings
 - A **fusion layer** to combine the output of the self-attention layers

$$\mathbf{h}_j = \sigma \left(\widetilde{\mathbf{W}}_t^{(i)} \widetilde{\mathbf{w}}_j^{(i)} + \widetilde{\mathbf{W}}_e^{(i)} \widetilde{\mathbf{e}}_k^{(i)} + \mathbf{b}^{(i)} \right) \quad \text{fusion representation}$$

$$\mathbf{w}_j^{(i)} = \sigma \left(\mathbf{W}_t^{(i)} \mathbf{h}_j + \mathbf{b}_t^{(i)} \right) \quad \text{token embedding output (fed to next block)}$$

$$\mathbf{e}_k^{(i)} = \sigma \left(\mathbf{W}_e^{(i)} \mathbf{h}_j + \mathbf{b}_e^{(i)} \right) \quad \text{entity embedding output (fed to next block)}$$

ERNIE: Enhanced Language Representation with Informative Entities [Zhang et al., ACL 2019]



ERNIE: Enhanced Language Representation with Informative Entities [Zhang et al., ACL 2019]

- How to train? Pretrain jointly with three tasks:
 - Masked language model and next sentence prediction (i.e., BERT tasks)
 - Knowledge pretraining task (dEA): randomly mask some token-entity alignments and predict which entity in the sequence should be linked to the given token

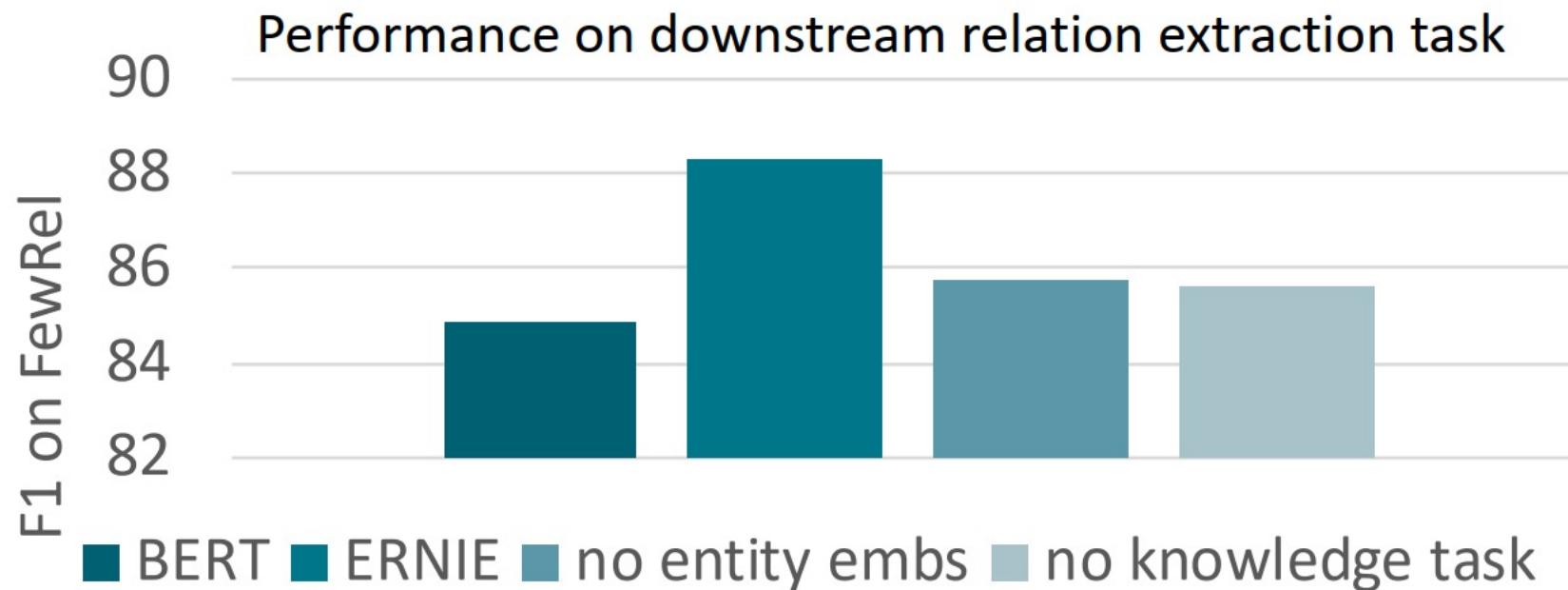
$$p(e_j | w_i) = \frac{\exp(\mathbf{W}w_i \cdot \mathbf{e}_j)}{\sum_{k=1}^m \exp(\mathbf{W}w_i \cdot \mathbf{e}_k)}$$

- Motivations: better learn word-entity alignments; and prevent overfitting to pre-given (ground-truth) entity linking inputs
- Final objective:

$$\mathcal{L}_{ERNIE} = \mathcal{L}_{MLM} + \mathcal{L}_{NSP} + \mathcal{L}_{dEA}$$

ERNIE: Enhanced Language Representation with Informative Entities [Zhang et al., ACL 2019]

- Analysis to see the effect of model components (entity embs and knowledge task)
 - Knowledge pretraining task is necessary to make the most use of the pretrained entity embeddings.



ERNIE: Enhanced Language Representation with Informative Entities [Zhang et al., ACL 2019]

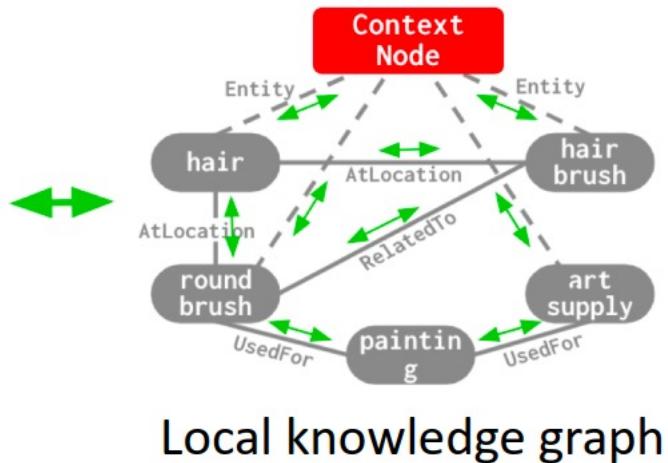
- Strengths:
 - Combines entity + text info through **fusion layers** and a **knowledge pretraining task**
 - Improves **performance** on knowledge-intensive downstream tasks
- Limitation:
 - Needs to **link each entity mention in input text to knowledge base** in advance
 - For instance, “Bob Dylan wrote Blowin’ in the Wind” needs entities linked to Wikidata knowledge base
 - It’s challenging to get a good entity linker for any domain of text or tasks
 - We will next talk about a more recent method that mitigates this issue

QAGNN/GreaseLM: Reasoning with Language Model and Knowledge Graph [Yasunaga et al. NAACL 2021; Zhang et al. ICLR 2022]

- Key idea: when adding entity embeddings to language model, **dynamically update** them together with neighbor or related entities in **knowledge graph** as well as text

[CTX] If it is not used for hair, a round brush is an example of what? Art supplies.

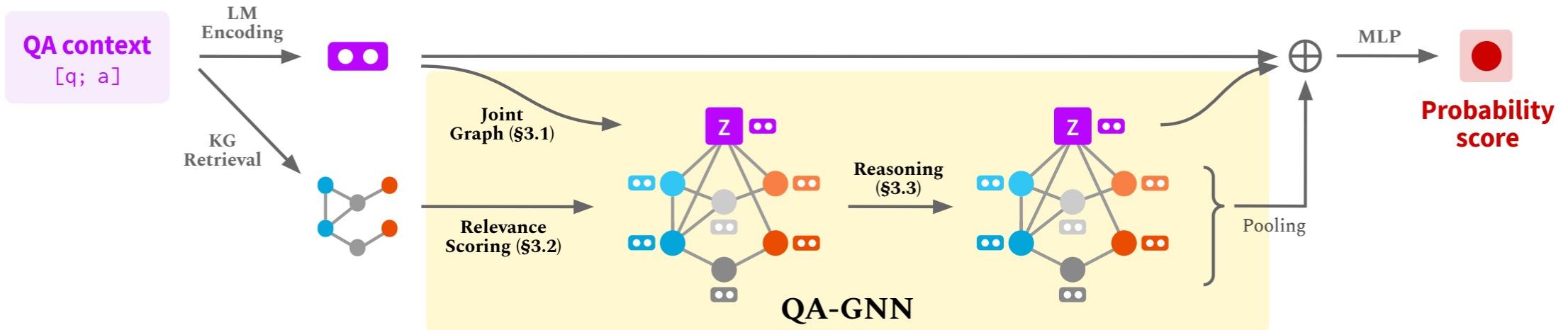
Text



Get all entity candidates and their neighbors in KG to prepare a local KG

- Benefits
 - **Robust** to non-perfect entity linking: can include all entity candidates and let the model figure out what to fuse
 - **Better contextualize knowledge**: helpful for joint reasoning about text and knowledge (e.g. question answering tasks)

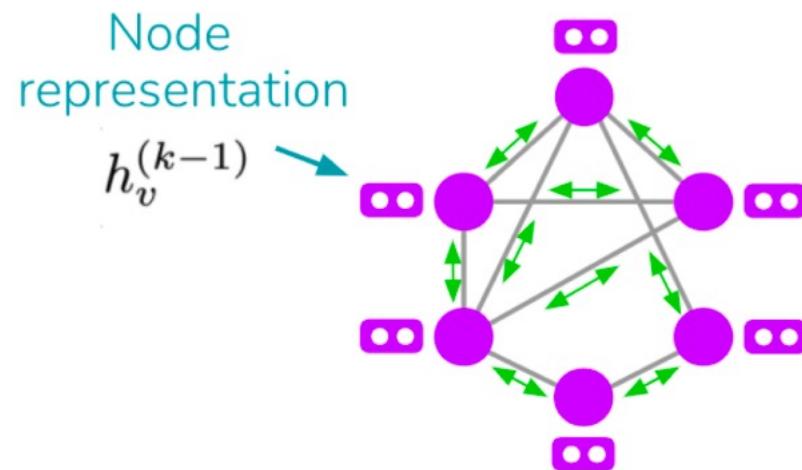
QAGNN



- Given a QA context (z), QAGNN 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).

QAGNN/GreaseLM: Reasoning with Language Model and Knowledge Graph [Yasunaga et al. NAACL 2021; Zhang et al. ICLR 2022]

- Model architecture:
- Text is encoded by a **language model**, knowledge graph (KG) is encoded by an **graph neural network (GNN)**, and they are fused together for multiple rounds



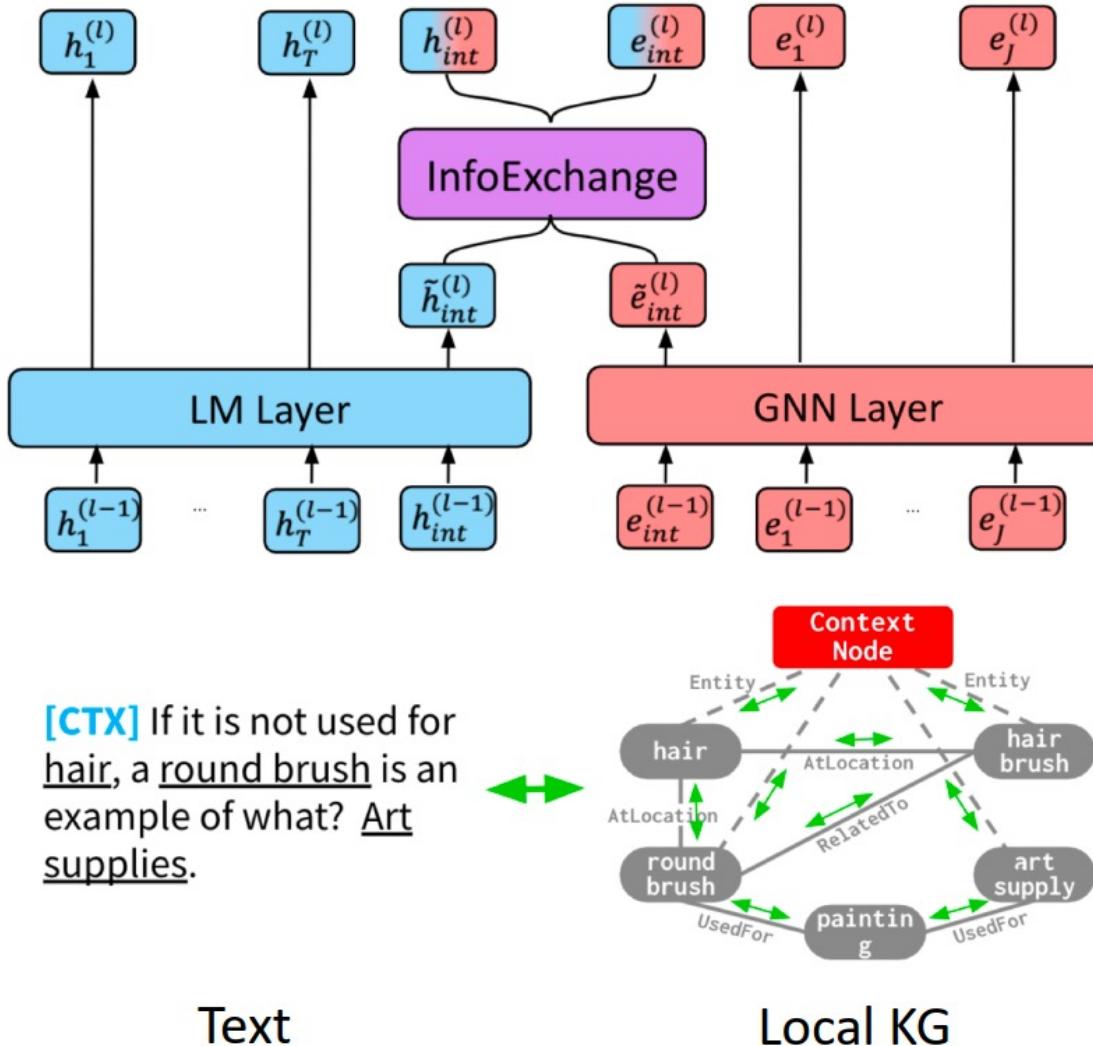
- What is GNN?
- Neural network designed for encoding graph data.
- GNN updates each node representation by aggregating message vectors from neighbor nodes.

$$a_v^{(k)} = \text{AGGREGATE}^{(k)} \left(\{ h_u^{(k-1)} : u \in \mathcal{N}(v) \} \right)$$

$$h_v^{(k)} = \text{COMBINE}^{(k)} \left(h_v^{(k-1)}, a_v^{(k)} \right)$$

QAGNN/GreaseLM: Reasoning with Language Model and Knowledge Graph [Yasunaga et al. NAACL 2021; Zhang et al. ICLR 2022]

- Model architecture:
- Text is encoded by a **language model**, knowledge graph (KG) is encoded by an **graph neural network (GNN)**, and they are fused together for multiple rounds



QAGNN/GreaseLM: Reasoning with Language Model and Knowledge Graph [Yasunaga et al. NAACL 2021; Zhang et al. ICLR 2022]

- Quantitative result: QAGNN and GreaseLM outperform previous BERT-based models on knowledge-intensive question answering tasks

Model	CommonsenseQA	OpenBookQA	MedQA
<u>BERT-Large</u>	55.4	60.4	-
<u>RoBERTa-Large</u>	68.7	64.8	35.0
<u>SapBERT-Base</u>	-	-	37.2
<u>QAGNN</u>	73.4	67.8	38.0
<u>GreaseLM</u>	74.2	66.9	38.5

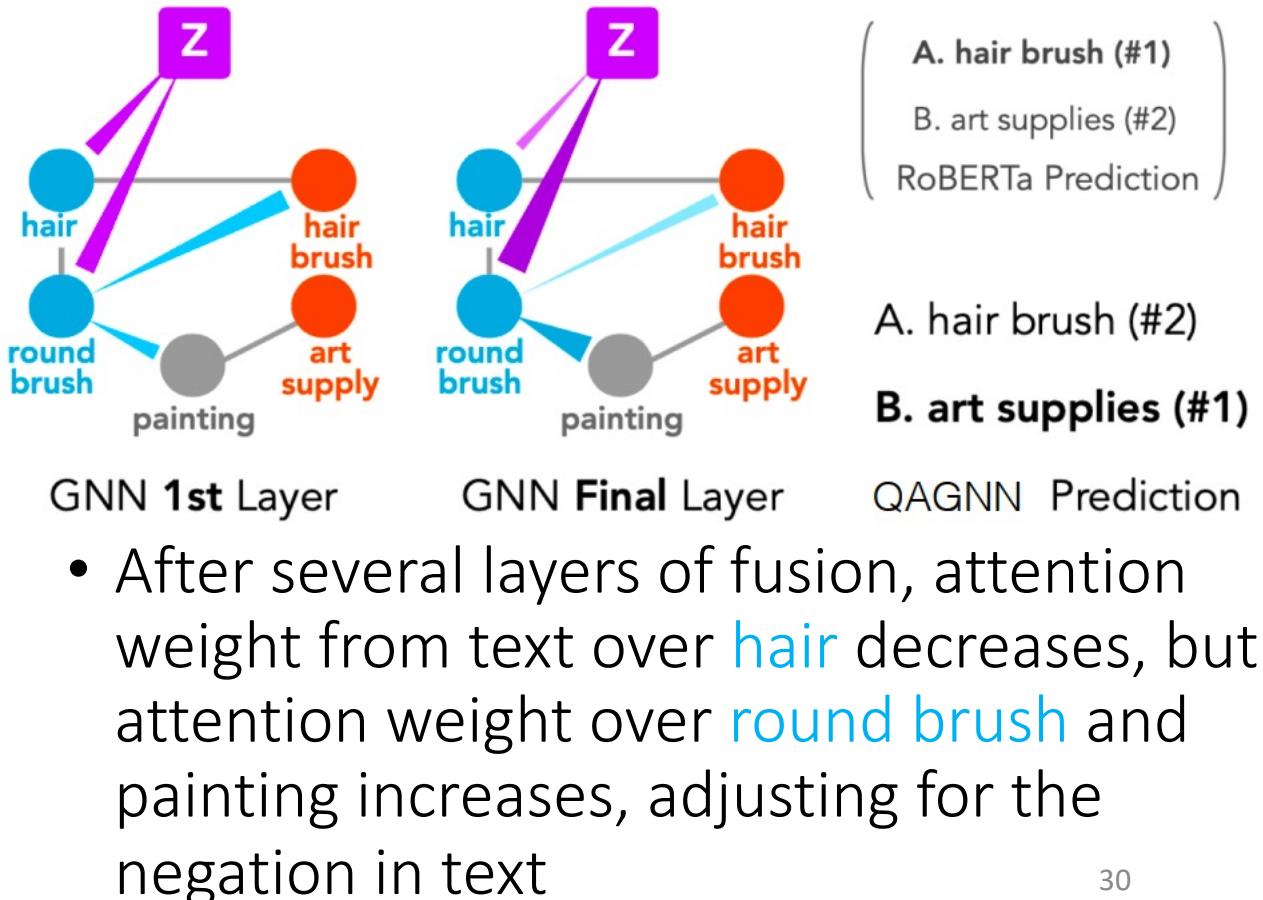
Devlin et al., NAACL 2019 & Liu et al., 2019 & Liu et al., NAACL 2021 & Yasunaga et al., NAACL 2021 & Zhang et al., ICLR 2022

QAGNN/GreaseLM: Reasoning with Language Model and Knowledge Graph [Yasunaga et al. NAACL 2021; Zhang et al. ICLR 2022]

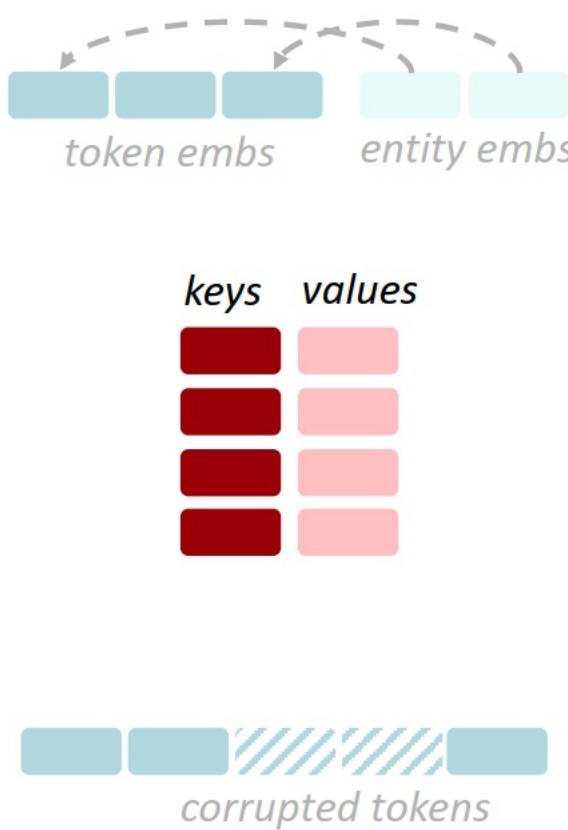
- Qualitative example:
- By grounding language model to knowledge graph, models learn to perform **structured reasoning** (e.g. handling negation correctly)
- Vanilla LMs don't handle negation well.. [Kassner et al., ACL 2020]
- New insight on how KG can help LM
 - Provide background knowledge
 - Provide **scaffold for reasoning**

If it is **not** used for **hair**, a **round brush** is an example of what?

- A. hair brush B. art supplies*



Techniques to add knowledge to LMs



- Add pretrained entity embeddings
 - ERNIE
 - QAGNN/GreaseLM
- Use an external memory
 - KGLM
- Modify the training data
 - WKLM
 - ERNIE (another!), salient span masking

Method 2: Use an external memory

- Previous methods rely on the **pretrained** entity embeddings to encode the factual knowledge into the language model.
 - Pros: Convenient, as you can just plug in any available entity embeddings
 - Cons: if the KB is modified, you may need to re-train the entity embeddings and model
- **Question:** Are there **more direct** ways to provide factual knowledge for LM?
- **Answer:** Yes! Give the model access to an external memory (a key-value store for KG triples or facts) in a way that is independent of learned model parameters
- **Advantages:**
 - Can directly update facts in the external memory without re-training the model
 - **Interpretable**
 - It's more visible which fact in external memory the LM used to make prediction ( it's harder to debug model predictions if we use entity embeddings)

External memory? → Memory Network

- 模型主要包含一系列的记忆单元（可以看成是一个数组，每个元素保存一句话的记忆）和 I, G, O, R 四个模块。

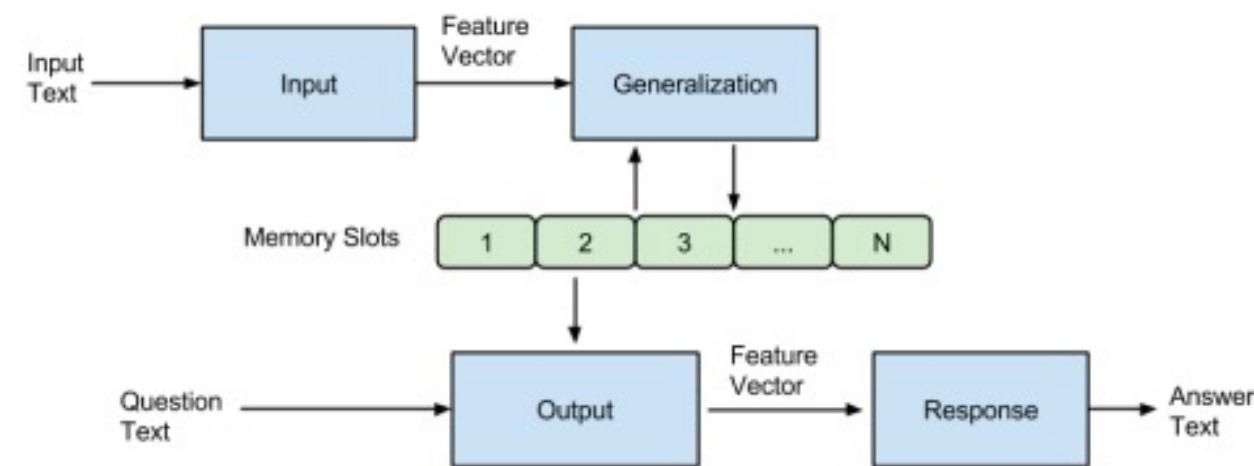
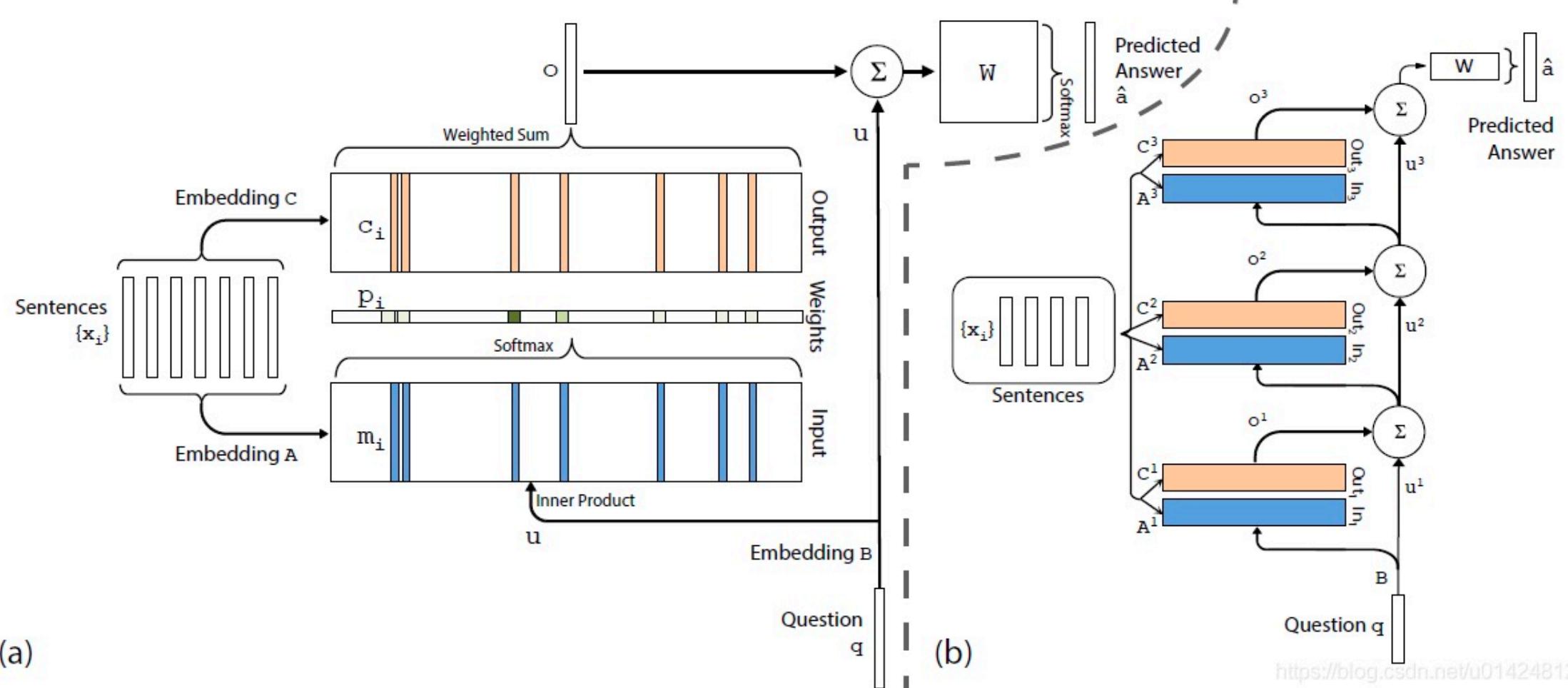


Figure 3: Memory Network architecture.

- I: (input feature map): 用于将输入转化为网络里内在的向量。（可以利用标准预处理，例如，文本输入的解析，共参考和实体解析。还可以将输入编码为内部特征表示，例如，从文本转换为稀疏或密集特征向量）
- G: (generalization): 更新记忆。在最初版本的实现里，只是简单地插入记忆数组里。也可以设计新的情况，包括了记忆的忘记，记忆的重新组织。（最简单的G形式是将 I (x) 存储在存储器中的“slot”中）
- O: (output feature map): 从记忆里结合输入，把合适的记忆抽取出来，返回一个向量。每次获得一个向量，代表了一次推理过程。
- R: (response): 将该向量转化回所需的格式，比如文字或者 answer。

End-To-End Memory Networks



<https://blog.csdn.net/u014248127>

Barack's Wife Hillary: Using Knowledge-Graphs for Fact-Aware Language Modeling (KGML) [Logan et al., ACL 2019]

- Key idea: **condition** the language model on a knowledge graph (KG) when predicting next word
- Recall that (standard) language models predict the next word given previous words:

$$P(x^{(t+1)} | x^{(t)}, \dots, x^{(1)})$$

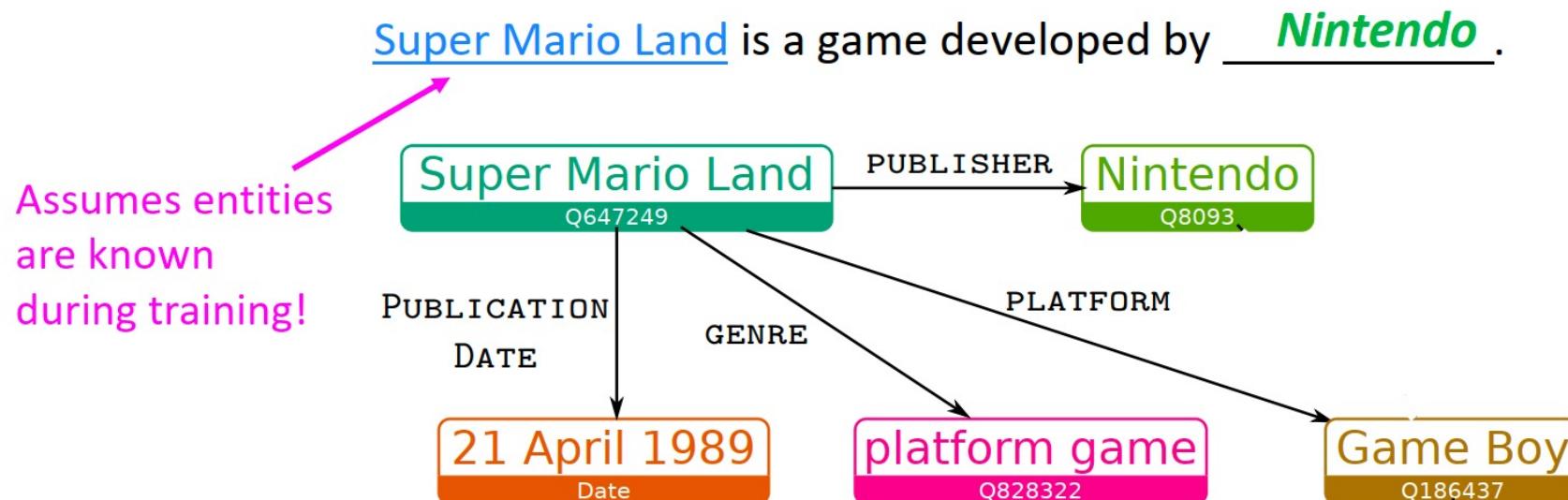
- Goal: predict the next word and entity using both the previous word and entity info

$$P(x^{(t+1)}, \mathcal{E}^{(t+1)} | x^{(t)}, \dots, x^{(1)}, \mathcal{E}^{(t)}, \dots, \mathcal{E}^{(1)})$$

- $\mathcal{E}^{(t)}$ is the set of KG entities mentioned at timestep

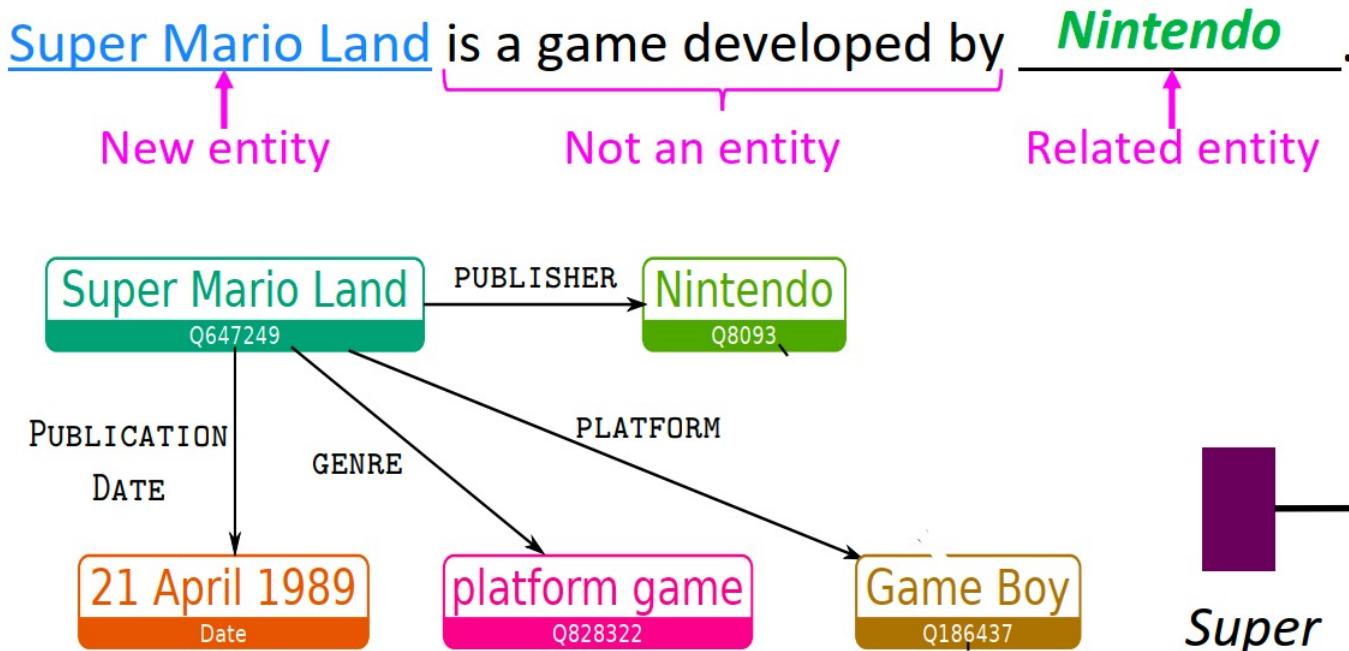
KGLM [Logan et al., ACL 2019]

- Method: Build a local knowledge graph as you iterate over the sequence
 - Local KG is a subset of the full KG with only entities relevant to the sequence so far



- Local KG can provide a strong signal for predicting what comes as the next word
- How can the LM know when to use the local KG vs standard LM to predict the next word?

KGLM [Logan et al., ACL 2019]



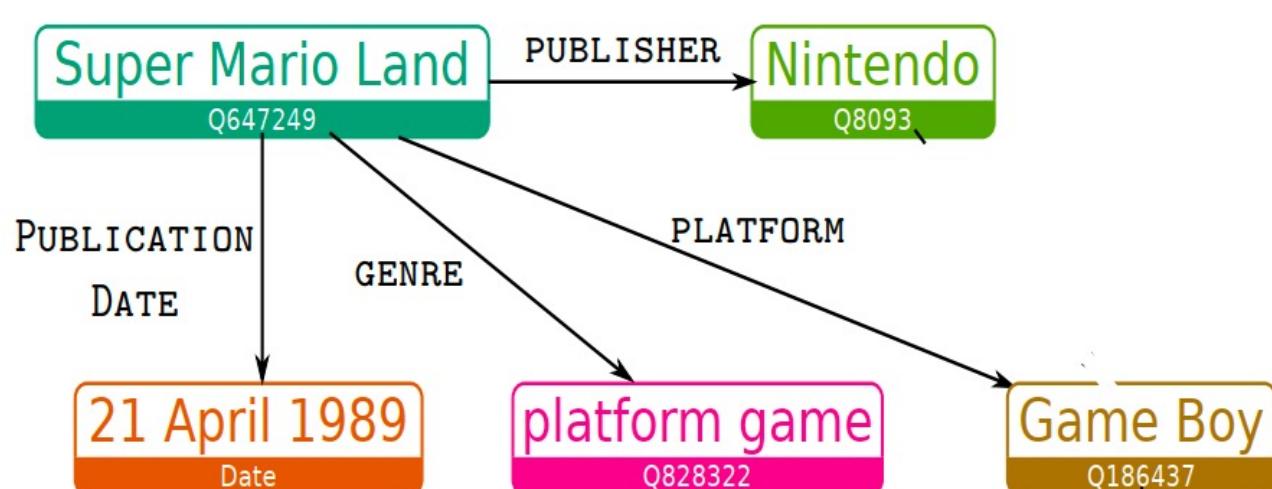
- Instead of predicting next word directly, use the LM hidden state to first predict the **type** of the next word (3 classes)
- Once we predict the word type, how to predict the next entity and word in each of the 3 scenarios?

KGLM [Logan et al., ACL 2019]

Super Mario Land is a game developed by Nintendo.

New entity Not an entity Related entity

- 1. Related entity case (in the local KG)



KG triple = (parent entity, relation, tail entity)

Example

Top scoring parent entity: “Super Mario Land”
Top scoring relation: “publisher”
-> Next entity is “Nintendo”, due to KG triple
(Super Mario Land, publisher, Nintendo).

KGLM [Logan et al., ACL 2019]

Super Mario Land is a game developed by Nintendo.

New entity Not an entity Related entity

- 1. Related entity case (in the local KG)
- Find the top-scoring parent and relation in the local KG using the LM hidden state and entity and relation embeddings
 - $P(p_t) = \text{softmax}(\mathbf{v}_p \cdot \mathbf{h}_t)$ where p_t is a potential parent entity, \mathbf{v}_p is the corresponding entity embedding, and \mathbf{h}_t is from the LM hidden state
 - Similarly for predicting top relation
- Next entity will be: tail entity from KG triple (top parent entity, top relation, tail entity)
- Next word will be: most likely next token over the standard vocabulary expanded to include the tail entity and its aliases¹

[1] Phrases that could all refer to Nintendo (e.g. Nintendo, Nintendo Co., Koppai)

KGLM [Logan et al., ACL 2019]

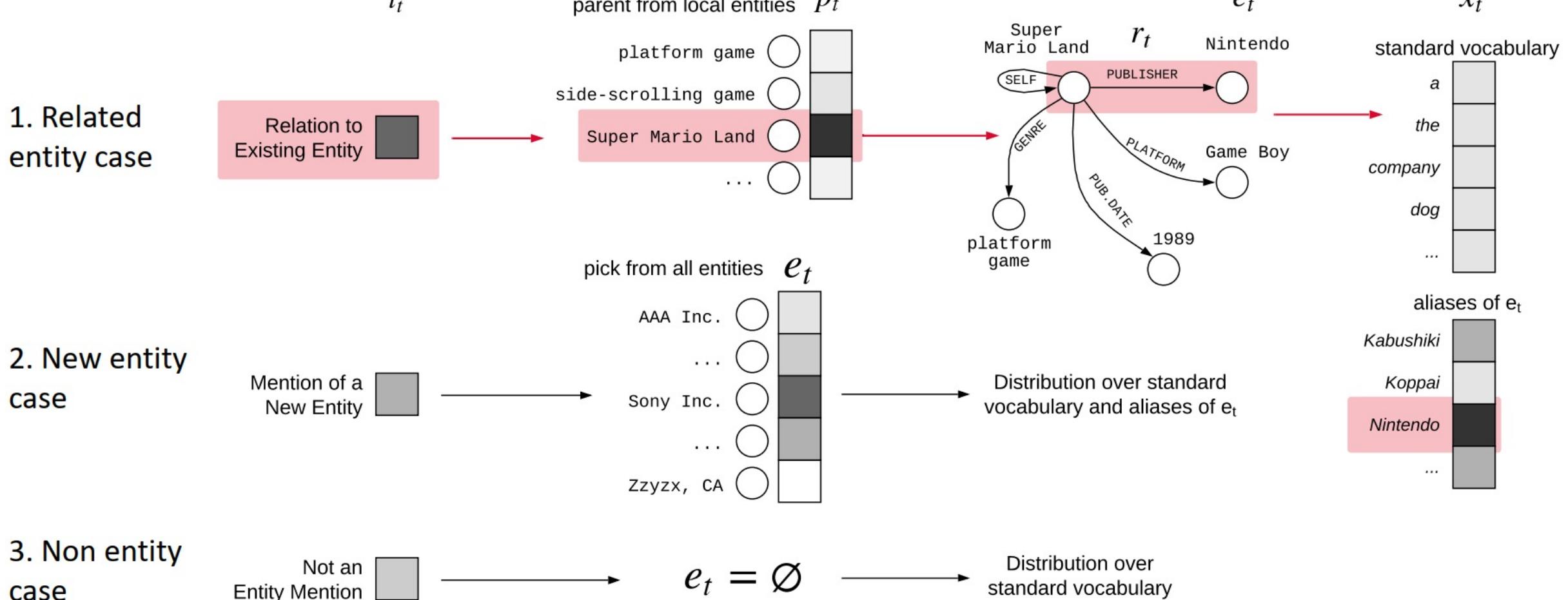
Super Mario Land is a game developed by Nintendo.

The diagram shows the sentence "Super Mario Land is a game developed by Nintendo." with three pink annotations. "Super Mario Land" is labeled "New entity" with an arrow pointing to it. "is a game developed by" is labeled "Not an entity" with a bracket underneath. "Nintendo" is labeled "Related entity" with an arrow pointing to it.

- 2. New entity case (not in the local KG)
 - Find the top-scoring entity in the full KG using the LM hidden state and entity embeddings
 - Next entity will be: the predicted top-scoring entity
 - Next word will be: most likely next token over standard vocabulary + entity aliases
- 3. Not an entity case
 - Next entity will be: None
 - Next word will be: most likely next token over standard vocabulary

KGLM [Logan et al., ACL 2019]

Super Mario Land is a 1989 side-scrolling platform video game developed and published by Nintendo

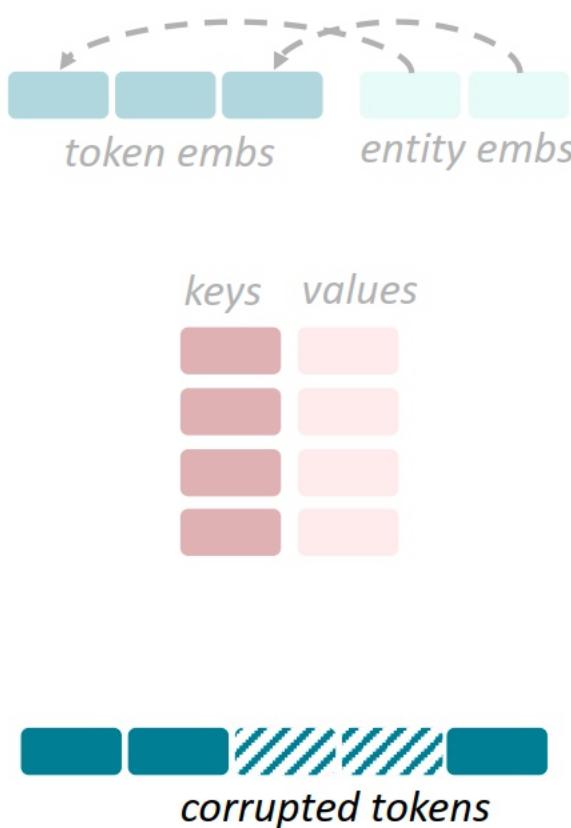


KGLM [Logan et al., ACL 2019]

- Outperforms GPT-2 and AWD-LSTM[Merity et al., ICLR 2018] on a fact completion task (“fill-in-the-blank”)
- Qualitatively, KGLM tends to **predict more specific tokens**, whereas GPT-2 predicts more common, generic tokens
- **Supports modifying/updating facts!**
 - Modifying the KG has a direct change in the LM predictions
 - Barack Obama was born on _____.
 - **KG triples:**
 - (Barack Obama, birthDate, 1961-08-04)
 - (Barack Obama, birthDate, **2013-03-21**)
- External memory can help LMs to do factually-grounded text generation!

Most likely next word:
“August”, “4”, “1961”
“March”, “21”, “2013”

Techniques to add knowledge to LMs



- Add pretrained entity embeddings
 - ERNIE
 - QAGNN/GreaseLM
- Use an external memory
 - KGLM
- Modify the training data
 - WKLM
 - ERNIE (another!), salient span masking

Method 3: Modify the training data

- Previous methods incorporated knowledge **explicitly** through pretrained embeddings and/or an external memory.
- Question: Can knowledge also be incorporated **implicitly** through the unstructured text?
- Answer: Yes! Mask or corrupt the data to introduce additional training tasks that require factual knowledge.
- **Advantages:**
 - No need for additional memory/computation (e.g. no need to carry a local KG)
 - No need for modifying the architecture (e.g. no need for a fusion layer)

Pretrained Encyclopedia: Weakly Supervised Knowledge Pretrained Language Model (WKLM) [Xiong et al., ICLR 2020]

- Key idea: train the model to distinguish between true and false knowledge
- Method: Replace mentions in the text with mentions that refer to different entities of the same type to create negative knowledge statements
 - Make the model predicts whether entity has been replaced or not
 - Need type-constraint to enforce linguistically correct replacement. Otherwise the model may trivially predict “replaced” using linguistic signal instead of knowledge

True knowledge statement:

J.K. Rowling is the author of Harry Potter.

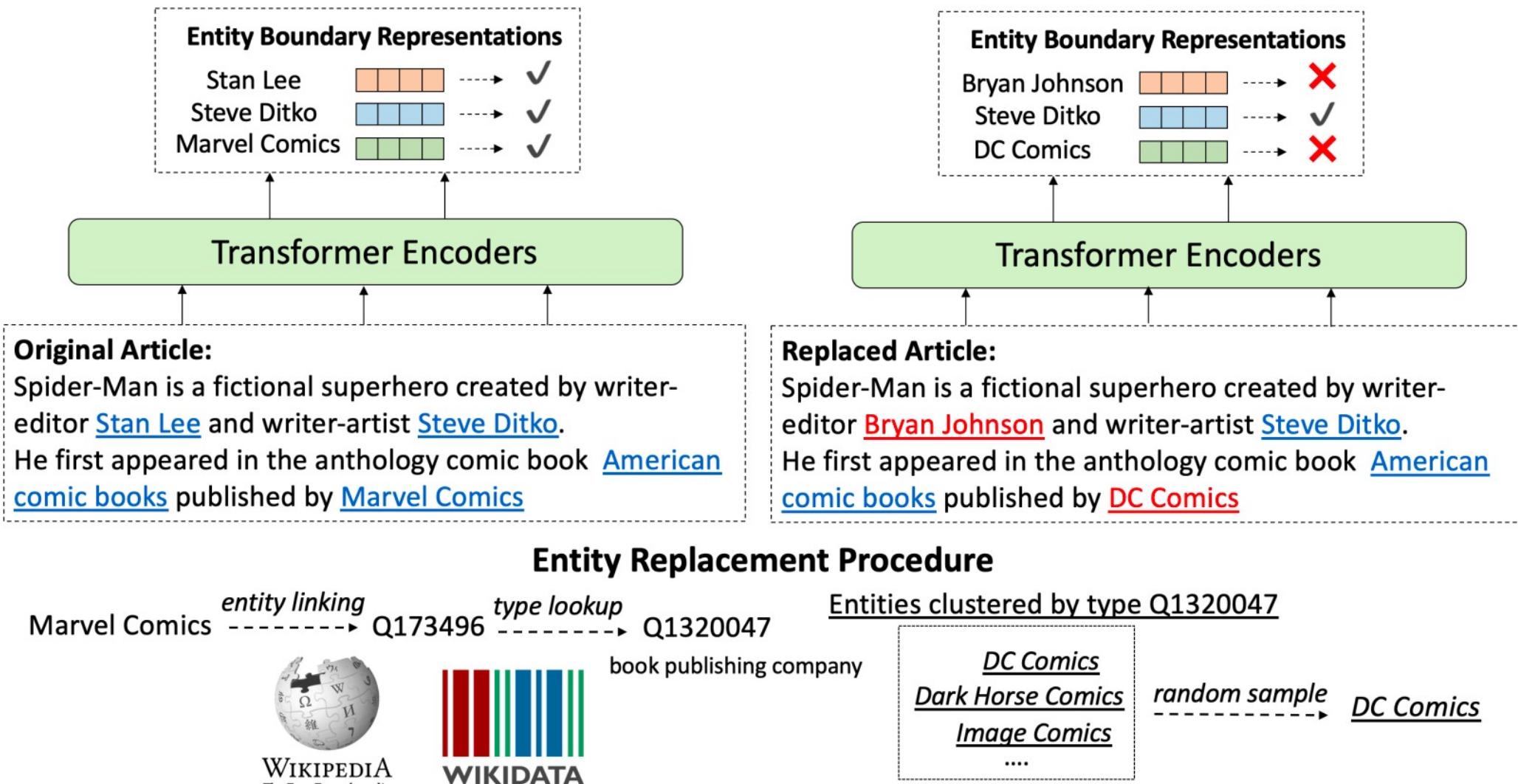


Negative knowledge statement:

J.R.R. Tolkien is the author of Harry Potter.

- => Requires the model to have background knowledge to be able to distinguish!

WKLM [Xiong et al., ICLR 2020]



WKLM [Xiong et al., ICLR 2020]

- **Training:** Uses an entity replacement loss (binary classification) to train the model to distinguish between true and false mentions

$$\mathcal{L}_{entRep} = \mathbb{I}_{e \in \mathcal{E}^+} \log P(e | C) + (1 - \mathbb{I}_{e \in \mathcal{E}^+}) \log(1 - P(e | C))$$

where e is an entity, C is the context, and \mathcal{E}^+ represents a true entity mention

- Total loss is the combination of standard masked language model loss (MLM) and the entity replacement loss.

$$\mathcal{L}_{WKLM} = \mathcal{L}_{MLM} + \mathcal{L}_{entRep}$$

- MLM is defined at the **token-level**; entRep is defined at the **entity-level**
 - Treating a **whole entity** (could be multi-word) instead of a token as **one unit** can make LMs more knowledge-aware

WKLM [Xiong et al., ICLR 2020]

- Improves over BERT and GPT-2 in fact completion tasks
- Improves over ERNIE on downstream tasks
- Ablation experiments (see the effect of model components, MLM and EntRep)
 - EntRep loss is essential because it makes WKLM outperform BERT
 - MLM loss is also essential for downstream task performance
 - On knowledge-intensive tasks, WKLM even outperforms training BERT longer with just MLM loss

Model	SQuAD (F1)	TriviaQA (F1)	Quasar-T (F1)	FIGER (acc)
WKLM	91.3	56.7	49.9	60.21
WKLM w/o MLM	87.6	52.5	48.1	58.44
BERT + 1M Updates	91.1	56.3	48.2	54.17

Much worse without MLM

Much worse training for longer, compared to using the entity replacement loss

Learn inductive biases through masking

- Besides corrupting data, another idea is: can we just do clever masking to help the LM learn factual knowledge?
 - ERNIE¹: Enhanced Representation through Knowledge Integration, Sun et al., arXiv 2019
 - Uses phrase-level and entity-level masking, and shows improvements on downstream NLP tasks
 - How Much Knowledge Can You Pack Into the Parameters of a Language Model?, Roberts et al., EMNLP 2020
 - Uses “salient span masking” (Guu et al., ICML 2020) to mask out salient spans (i.e. named entities and dates)
 - Shows that salient span masking improves T5’s performance on QA tasks

另外一个
ERNIE

Another ERNIE from Baidu

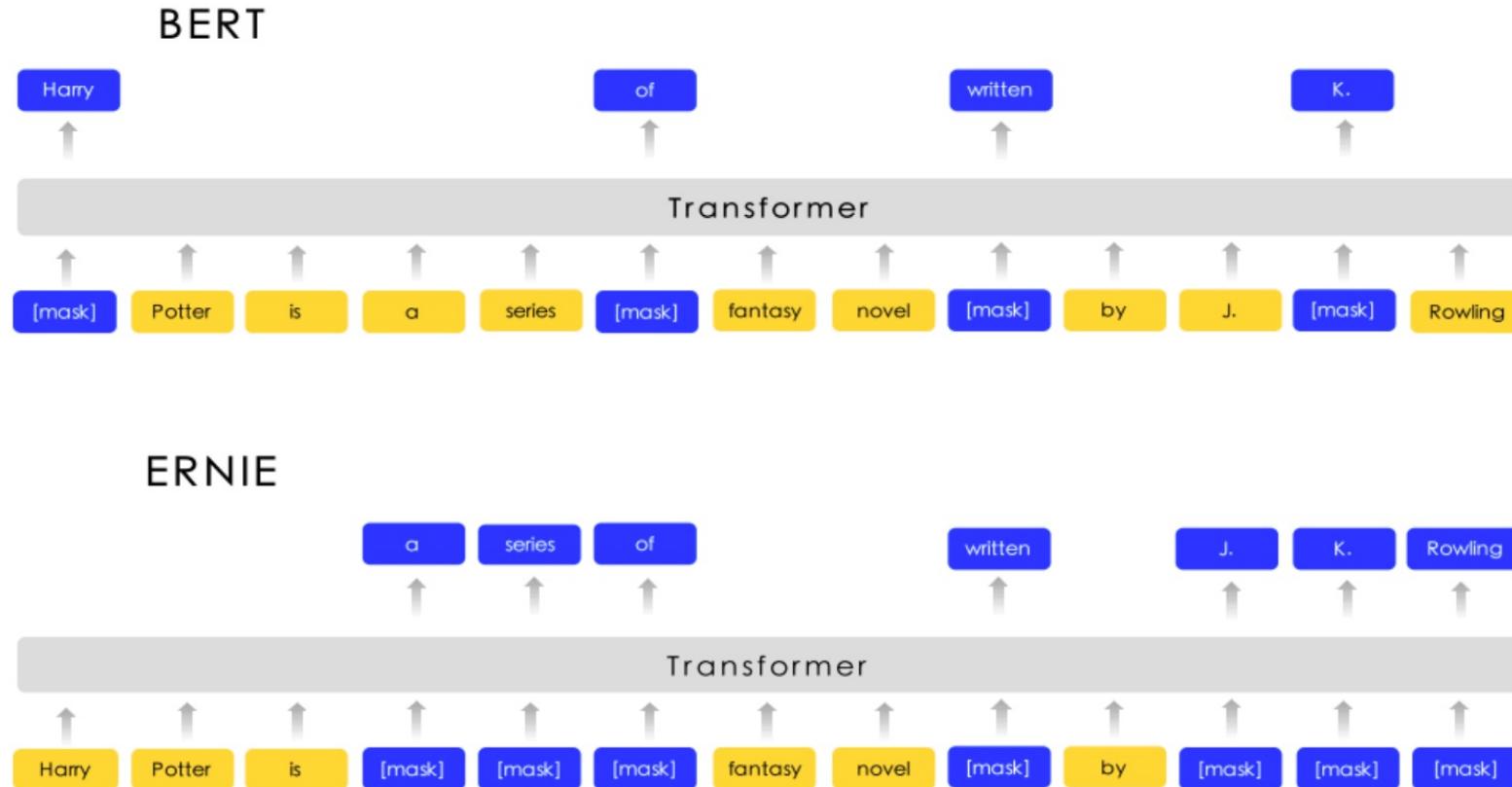
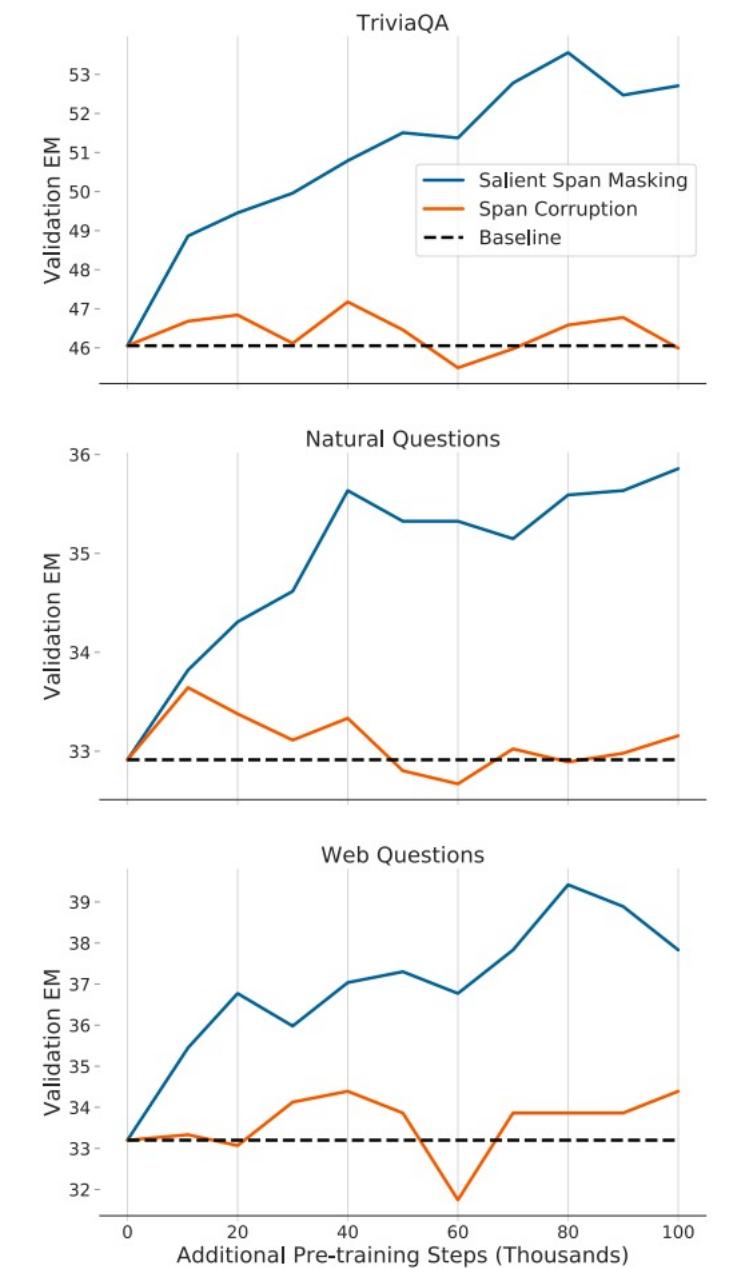


Figure 1: The different masking strategy between BERT and ERNIE

Salient span masking

- Salient span masking has been shown to outperform other masking or corruption strategies on QA and document retrieval tasks.
- QA/Retrieval performance on NaturalQuestions

Masking technique	Exact Match	Retrieval Recall @5
<u>Random uniform masks</u> (BERT)	32.3	24.2
<u>Random span masks</u> (SpanBERT)	35.3	26.1
Salient span masking	38.2	38.5



Evaluating knowledge in LMs

- Probes
- Downstream tasks

Probes

LAnguage Model Analysis (LAMA) Probe [Petroni et al., EMNLP 2019]

- Idea: How much relational (**commonsense** and **factual**) knowledge is already in off-the-shelf language models?
 - Without any additional training or fine-tuning
- Manually constructed a set of “**cloze**” statements (**fill-in-the-blank**) to assess a model’s ability to predict a missing token.
- Examples:
 - The theory of relativity was developed by [MASK].
 - The native language of Mammootty is [MASK].
 - Ravens can [MASK].
 - You are likely to find a overflow in a [MASK].



LAnguage Model Analysis (LAMA) Probe [Petroni et al., EMNLP 2019]

- Generate cloze statements from KG triples and question-answer pairs in QA datasets
- Goal: evaluate knowledge in off-the-shelf pretrained LMs (Note: this means they may have used different pretraining corpora)
- Compare the unsupervised LMs to supervised relation extraction (RE) and QA systems

Mean precision at one (P@1)

Corpus	DrQA	RE baseline	ELMo	ELMo (5.5B)	GPT2 (300M)	GPT2 (1.5B)	BERT-base	BERT-large
Google-RE	-	7.6	2.0	3.0	4.9	6.5	9.8	10.5
T-REx	-	33.8	4.7	7.1	20.3	25.1	31.1	32.2
ConceptNet	-	-	6.1	6.2	9.7	12.8	15.6	19.2
SQuAD	37.5	-	1.6	4.3	5.9	11.5	14.1	17.4

BERT struggles on N-to-M relations

LMs are NOT finetuned!

LAnguage Model Analysis (LAMA) Probe [Petroni et al., EMNLP 2019]

- Using LAMA library, you can try out examples to assess knowledge in popular/your favorite LMs!
- <https://github.com/facebookresearch/LAMA>
- The cat is on the [MASK]

bert:					
Top10 predictions					
0	phone	-2.345			
1	floor	-2.630			
2	ground	-2.968			
3	couch	-3.387			
4	move	-3.649			
5	roof	-3.651			
6	way	-3.718			
7	run	-3.757			
8	bed	-3.802			
9	left	-3.965			

index	token	log_prob	prediction	log_prob	rank@1000

1	The	-5.547	.	-0.607	14
2	cat	-0.367	cat	-0.367	0
3	is	-0.019	is	-0.019	0
4	on	-0.001	on	-0.001	0
5	the	-0.002	the	-0.002	0
6	[MASK]	-14.321	phone	-2.345	-1
7	.	-0.002	.	-0.002	0



LAnguage Model Analysis (LAMA) Probe [Petroni et al., EMNLP 2019]

- Limitations of the LAMA probe:
 - Hard to understand *why* models perform well when they do
 - LM could just be memorizing word co-occurrence patterns rather than “understanding” the cloze statement and “recalling” knowledge
 - LM could just be identifying similarities between the surface forms of the subject and object (e.g., Pope Clement VII has the position of pope)
 - LMs are sensitive to the phrasing of the statement
 - e.g. sometimes rephrasing the template makes LMs suddenly perform better
 - But LAMA has only one manually defined template for each relation
 - This means probe results are a *lower bound* on knowledge encoded in the LM
- We will talk about two works that address these limitations

LAnguage Model Analysis (LAMA) Probe [Petroni et al., EMNLP 2019]

- Limitations of the LAMA probe:
 - Hard to understand **why** models perform well when they do
 - LM could just be memorizing word co-occurrence patterns rather than “understanding” the cloze statement and “recalling” knowledge
 - LM could just be identifying similarities between the surface forms of the subject and object (e.g., Pope Clement VII has the position of pope)
 - LMs are sensitive to the phrasing of the statement
 - e.g. sometimes rephrasing the template makes LMs suddenly perform better
 - But LAMA has only one manually defined template for each relation
 - This means probe results are a lower bound on knowledge encoded in the LM
- We will talk about two works that address these limitations

A More Challenging Probe: LAMA-UnHelpful Names (LAMA-UHN) [Poerner et al., EMNLP 2020]

- Key idea: Remove the examples from LAMA that can be answered **without relational knowledge**
- Motivation: BERT may rely on surface forms of entities to make predictions
 - String match between subject and object
 - “Revealing” person name: Name can be a (possibly incorrect) prior for native language, nationality, etc.
- Removing these examples helps to evaluate whether BERT is really knowing the fact
- With LAMA-UHN, BERT’s score drops ~8%
 - Knowledge-enhanced model E-BERT drops only <1%

Native language of French-speaking actors according to BERT

Person Name	BERT
Jean Marais	French
Daniel Ceccaldi	Italian
Orane Demazis	Albanian
Sylvia Lopez	Spanish
Annick Alane	English

LAnguage Model Analysis (LAMA) Probe [Petroni et al., EMNLP 2019]

- Limitations of the LAMA probe:
 - Hard to understand why models perform well when they do
 - LM could just be memorizing word co-occurrence patterns rather than “understanding” the cloze statement and “recalling” knowledge
 - LM could just be identifying similarities between the surface forms of the subject and object (e.g., Pope Clement VII has the position of pope)
 - LMs are sensitive to the phrasing of the statement
 - e.g. sometimes rephrasing the template makes LMs suddenly perform better
 - But LAMA has only one manually defined template for each relation
 - This means probe results are a **lower bound** on knowledge encoded in the LM
- We will talk about two works that address these limitations

Developing better prompts to query knowledge in LMs

[Jiang et al., TACL 2020]

- Problem: LMs may know the fact, but fail on completion tasks (LAMA) due to the query phrasing
 - Pretraining text may have had different sentence structures/contexts than the query
 - Example: “The birth place of Barack Obama is Honolulu, Hawaii” (pretraining corpus) versus “Barack Obama was born in _____” (query)
- Solution
 - Generate more LAMA prompts by mining templates from Wikipedia and generating paraphrased prompts by using back-translation
 - Increases the chance of getting a prompt similar to what was seen in pretraining
 - Ensemble prompts: LM’s output probability is averaged over different prompts

Developing better prompts to query knowledge in LMs

[Jiang et al., TACL 2020]

- Results: Performance on LAMA for BERT-large increases 7% when using top-performing query for each relation. Ensembling leads to another 4% gain.
 - Original LAMA really was a lower bound on knowledge encoded in LM!
- Small changes in the query phrasing lead to large gains.
 - LMs are very sensitive to the query phrasing => research opportunity for robust LM!

ID	Modifications	Acc. Gain
P413	x plays in → at y position	+23.2
P495	x was created → made in y	+10.8
P495	x was → is created in y	+10.0
P361	x is a part of y	+2.7
P413	x plays in y position	+2.2

Downstream tasks

Knowledge-intensive downstream tasks

- Measures how well the knowledge-enhanced LM transfers its knowledge to downstream tasks
- Unlike probes, this evaluation usually involves finetuning the LM on downstream tasks, like evaluating BERT on GLUE tasks
- Common knowledge-intensive tasks:
 - Relation extraction
 - Example: [Bill Gates] was born in [Seattle]; label: “city of birth”
 - Entity typing
 - Example: [Alice] has donated billions to eradicate malaria; label: “philanthropist”
 - Question answering
 - Example: “What kind of forest is the Amazon?”; label: “moist broadleaf forest”

Relation extraction performance on TACRED

- Knowledge-enhanced systems (ERNIE, KnowBERT) improve over previously state-of-the-art models for relation extraction

Model	LM	Precision	Recall	F1
C-GCN	-	69.9	63.3	66.4
BERT-LSTM-base	BERT-Base	73.3	63.1	67.8
ERNIE (Zhang et al.)	BERT-Base	70.0	66.1	68.0
KnowBert-W+W	BERT-Base	71.6	71.4	71.5

Peters et al., EMNLP 2019

Entity typing performance on OpenEntity

- Knowledge-enhanced LMs (ERNIE, KnowBERT) improve over prior LSTM and BERT-Base models on entity typing
- Impressively, previous models (NFGEC, UFET) were designed for entity typing

Model	Precision	Recall	F1
<u>NFGEC</u> (LSTM)	68.8	53.3	60.1
<u>UFET</u> (LSTM)	77.4	60.6	68.0
<u>BERT-Base</u>	76.4	71.0	73.6
<u>ERNIE</u> (Zhang et al.)	78.4	72.9	75.6
<u>KnowBert-W+W</u>	78.6	73.7	76.1

Zhang et al., ACL 2019 & Peters et al., EMNLP 2019

Knowledge-intensive Question Answering

- Knowledge-enhanced LMs (QAGNN, GreaseLM) improve over previous BERT-based models on question answering

Model	CommonsenseQA	OpenBookQA	MedQA
<u>BERT-Large</u>	55.4	60.4	-
<u>RoBERTa-Large</u>	68.7	64.8	35.0
<u>SapBERT-Base</u>	-	-	37.2
<u>QAGNN</u>	73.4	67.8	38.0
<u>GreaseLM</u>	74.2	66.9	38.5

Devlin et al., NAACL 2019 & Liu et al., 2019 & Liu et al., NAACL 2021 & Yasunaga et al., NAACL 2021 & Zhang et al., ICLR 2022

Summary of Evaluation

- Probes
 - Evaluate the knowledge already present in models [without more training](#)
 - Challenging to construct benchmarks that really test factual knowledge
 - Challenging to construct the query prompts used in the probe
- Downstream tasks
 - Evaluate the [usefulness](#) of the knowledge-enhanced representation [in applications](#)
 - Typically requires finetuning the LM further on the downstream task
 - Less direct way to evaluate the knowledge in the LM, but perhaps more practically useful in terms of applications

Other exciting progress!

- Retrieval-augmented language models
 - REALM, Guu et al., ICML 2020
 - RAG, Lewis et al., NeurIPS 2020
 - Retro, Borgeaud et al., 2022
- Modifying knowledge in language models
 - Fast Model Editing at Scale, Mitchell et al., 2021
- More knowledge-aware pretraining for language models
 - KEPLER, Wang et al., TACL 2020
- More efficient knowledge systems
 - NeurIPS Efficient QA challenge
- Better knowledge benchmarks
 - KILT, Petroni et al., NAACL 2021

Thank you