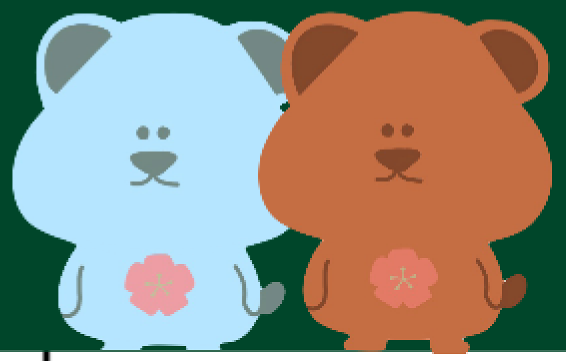
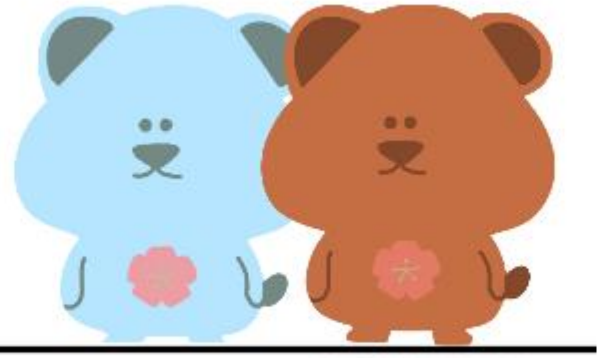


Integrating knowledge in Language Models

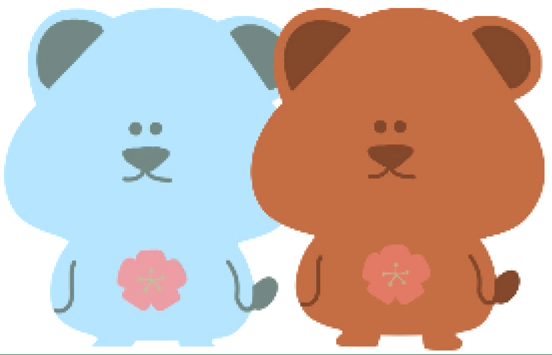
송혜준 주연우

EWHA EURON

목차



#01 What does a LM know?



#01 What does a LM know?

Recap: LM

- Standard language models predict the next word in a sequence of text and can compute the probability of a sequence

The students opened their **books** *.*

- Recently, masked language models (e.g., BERT) instead predict a masked token in a sequence of text using bidirectional context

went **store**
I [MASK] to the [MASK].

- Both types of language models can be trained over large amounts of unlabeled text!

#01 What does a LM know?

Recap: LM

- Traditionally, LMs are used for many tasks involving **generating** or **evaluating** the probability of text:

- Today, LMs are commonly used to generate **pretrained representations** of text that encode some notion of language understanding for downstream NLP tasks

- Can a language model be used as a **knowledge base**?

#01 What does a LM know?

1. Why the predictions are not all factually correct?

Predictions generally make sense, 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 example during training to memorize the fact
  - **Model sensitivity** : LM may have seen the fact during training, but is sensitive to the phrasing
    - Correctly answers "x was <u>made</u> in y" templates but not "x was <u>created</u> in y"

- The inability to "**reliably**" recall knowledge is a key challenge facing LMs today

#01 What does a LM know?

2. The importance of knowledge-aware LM

LM pretrained representations can benefit downstream tasks that leverage knowledge

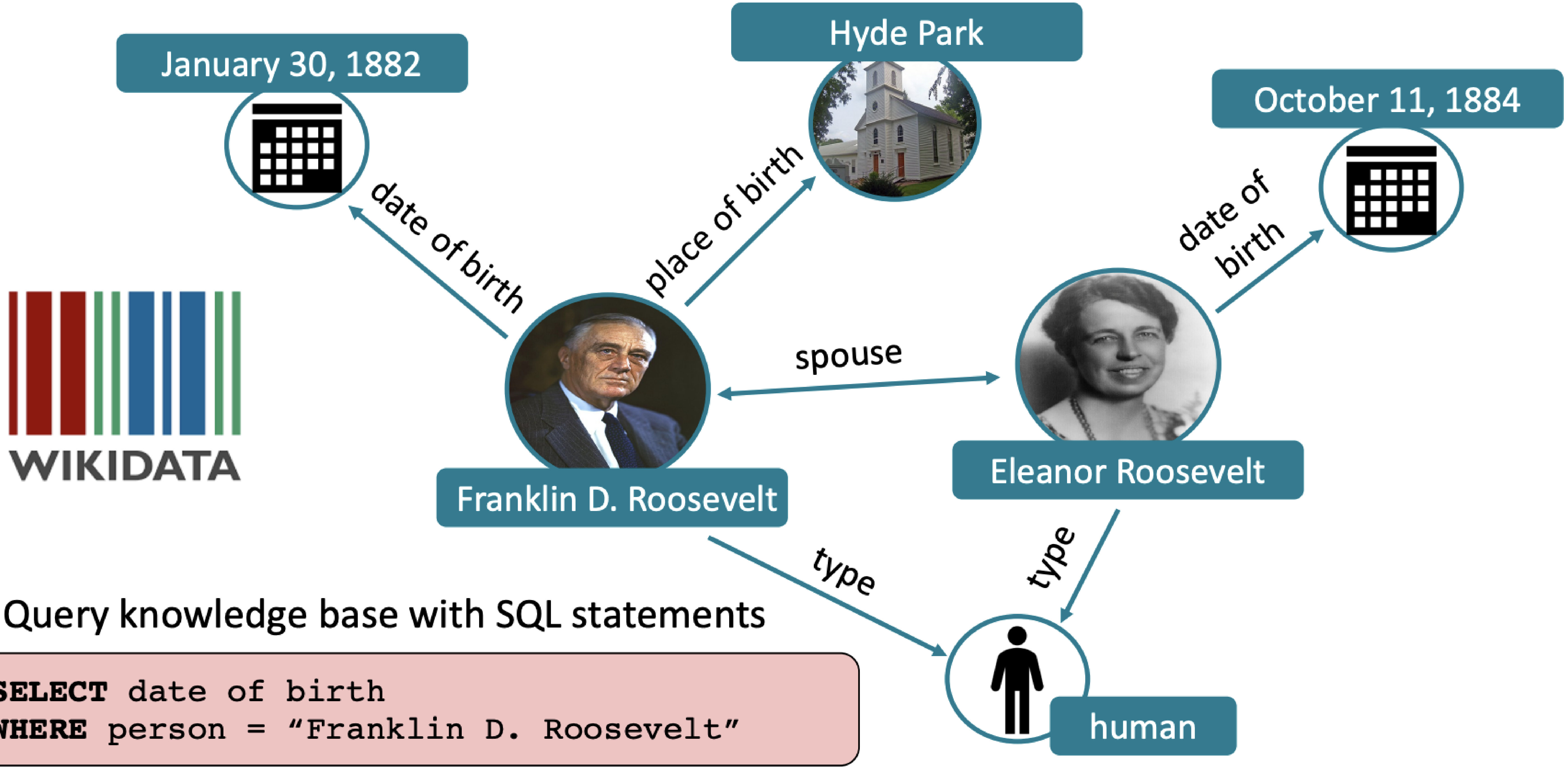
- For instance, extracting the relations between two entities in a sentence is easier with some knowledge of the entities

Stretch goal: Can LMs ultimately replace traditional knowledge bases?

- Instead of querying a knowledge base for a fact (e.g. with SQL), query the LM with a natural language prompt!
- Of course, this requires LM to have high quality on recalling facts

#01 What does a LM know?

3. Querying traditional knowledge bases



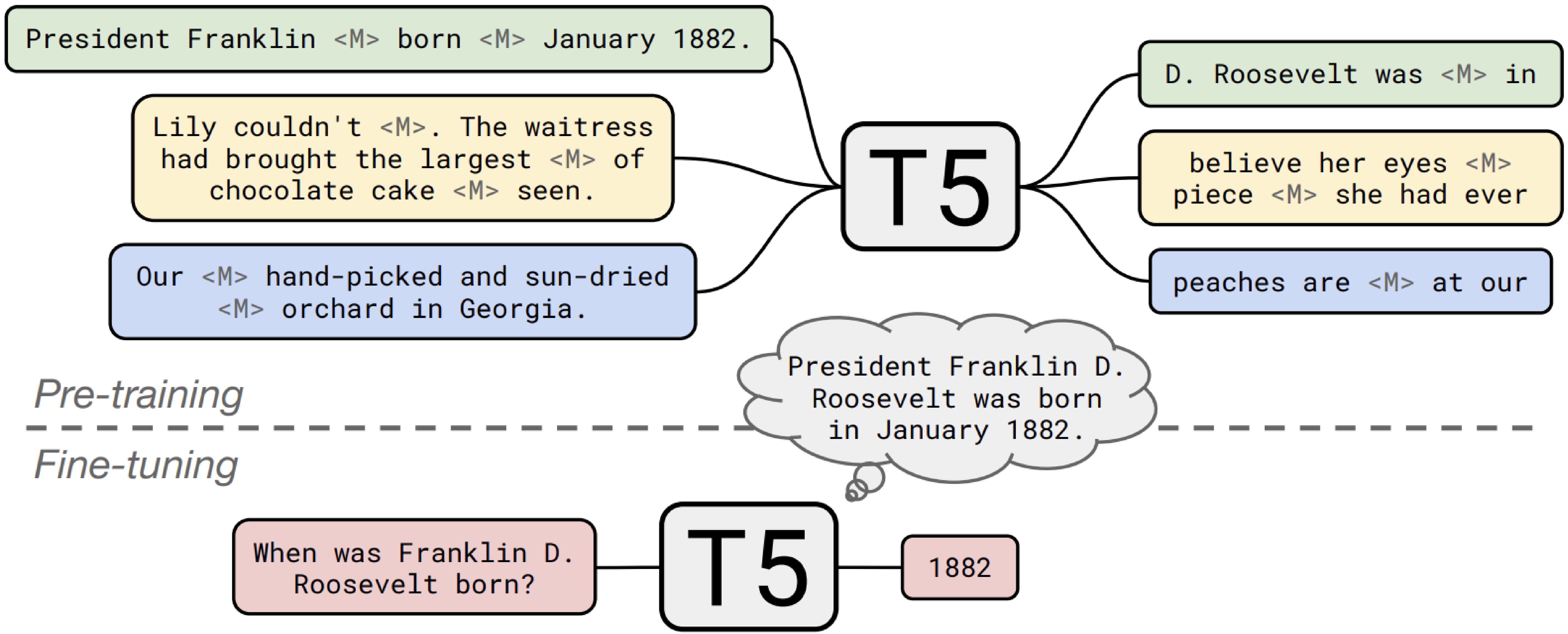
- Query knowledge base with SQL statements

```
SELECT date of birth
WHERE person = "Franklin D. Roosevelt"
```

#01 What does a LM know?

4. Querying language models as knowledge bases

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



Cs224n_slides

#01 What does a LM know?

Comparison between knowledge Graph and LMs as knowledge bases

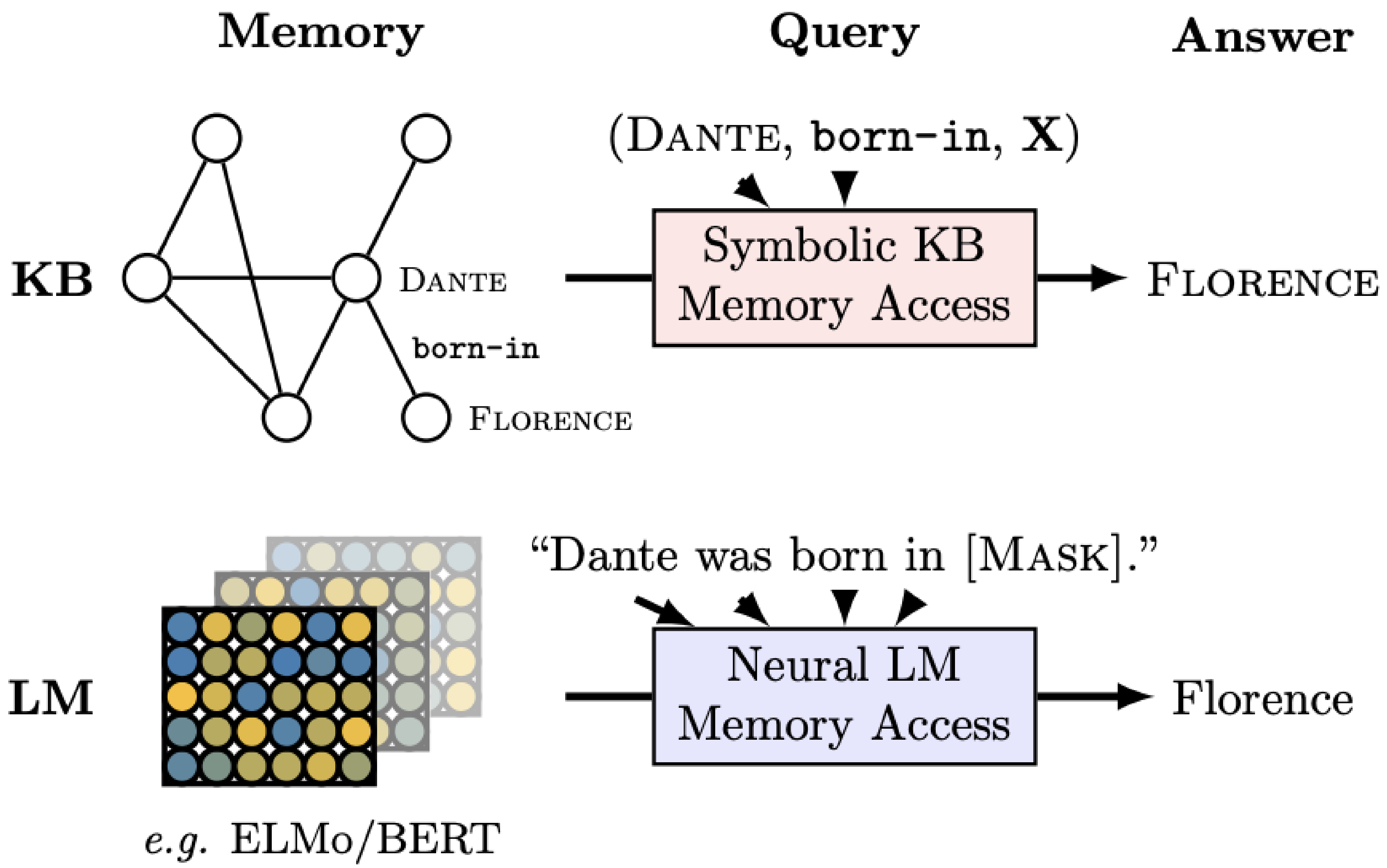


Figure 1: Querying knowledge bases (KB) and language models (LM) for factual knowledge.

https://aclanthology.org/D19-1250.pdf

#01 What does a LM know?

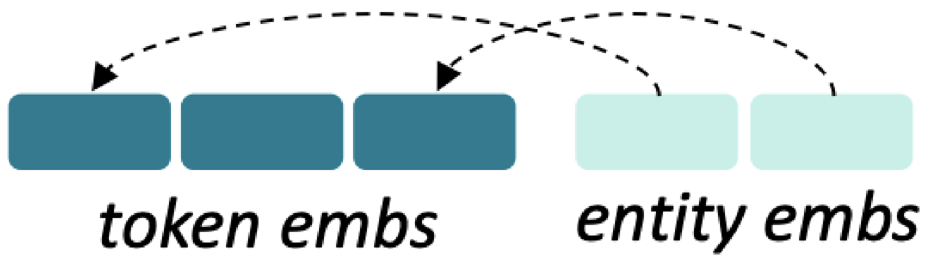
5. Advantages of language models over traditional KBs

- LMs are pretrained over large amounts of unstructured and unlabeled text
  - KBs require manual annotation or complex NLP pipelines to populate
- LMs support more flexible natural language queries
  - Example: What does the final F in the song U.F.O.F. stand for?
    - Traditional KB wouldn't have a field for "final F"; LM may learn this
- However, there are also many open challenges to using LMs as KBs:
  - Hard to interpret (i.e., why does the LM produce an answer)
  - Hard to trust (i.e., the LM may produce a realistic, incorrect answer)
  - Hard to modify (i.e., not easy to remove or update knowledge in the LM)

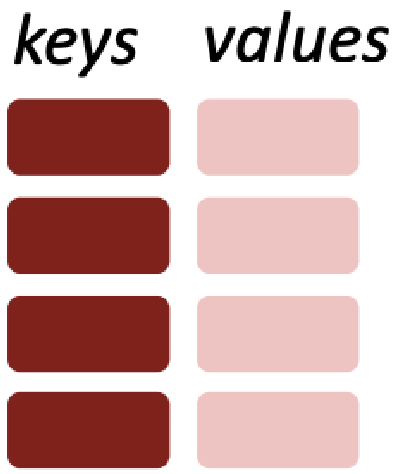
#02 Techniques to add knowledge to LMs

#02 Techniques to add knowledge to LMs

Techniques to add knowledge to LMs



1. Add pretrained entity embeddings
   - ERNIE



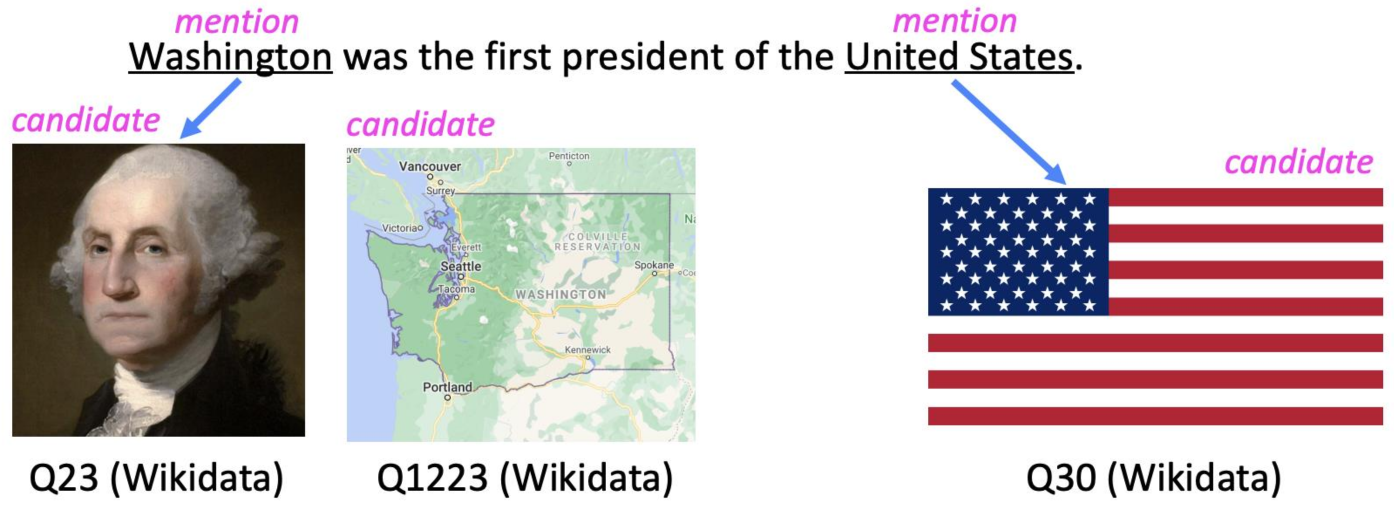
2. Use an external memory
   - KGLM
   - kNN-LM

3. Modify the training data
   - WKLM
   - ERNIE (another!), salient span masking

#02 Techniques to add knowledge to LMs

1. Add pretrained entity embeddings

- Facts about the world are usually in terms of entities
  - example: Washington was the first president of the United States.
- Pretrained word embeddings do not have a notion of entities
  - Different word embeddings for “U.S.A.”, “United States of America” and “America” even though these refer to the same entity
- What if we assign an embedding per entity?
  - ==Single entity embedding== for “U.S.A.”, “United States of America” and “America”
- Entity embeddings can be useful to LMs iff you can do ==entity linking== well!



- Like mentions in text to entities in a knowledge base
- Entity linking tells us which entity embeddings are relevant to the text

#02 Techniques to add knowledge to LMs

1. Add pretrained entity embeddings

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

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)

- knowledge를 LM에 추가할 때 text로 표현된 entity를 embedding해서 LM이 언어를 이해할 수 있게 해야 함
- Entity embedding 방식 중 하나인 TransE (ERNIE 모델에서 사용) 를 이해하는 것 중요함
- Knowledge graph embedding이란?

Cs224n_slides

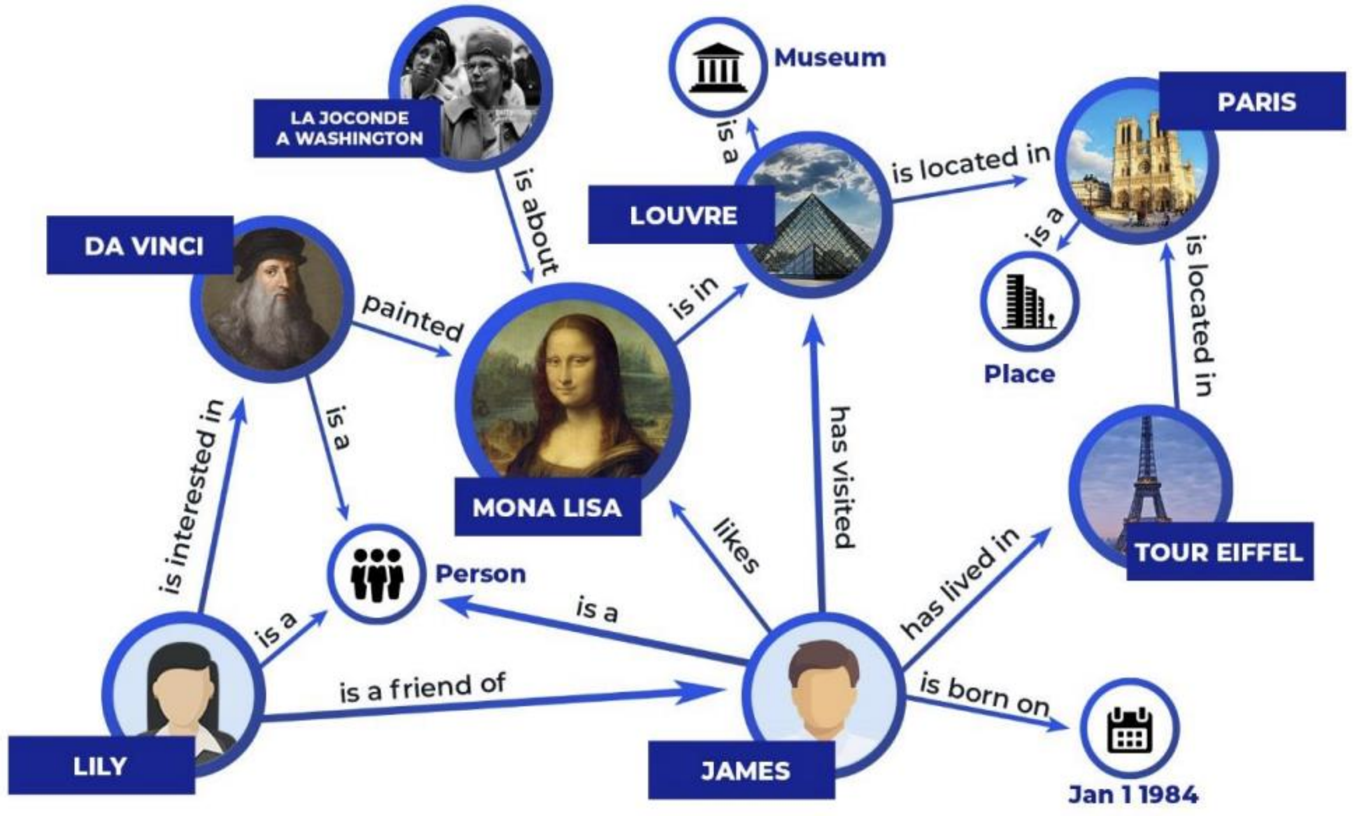
#02 Techniques to add knowledge to LMs

Asides: Knowledge Graph (KG)

- Knowledge Base (KB)
  - 현실의 지식을 저장한 대규모 데이터베이스
  - Wikidata, DBPedia
- Knowledge Graph의 형식으로 지식 저장

- Knowledge Graph
  - 객체(Entity)들 간의 관계(Relation)가 표현된 방향을 가진 그래프
  - Node: Entity(고유명사, 연도, 대표속성 등)
  - Edge: Relation(관계)

Knowledge Base

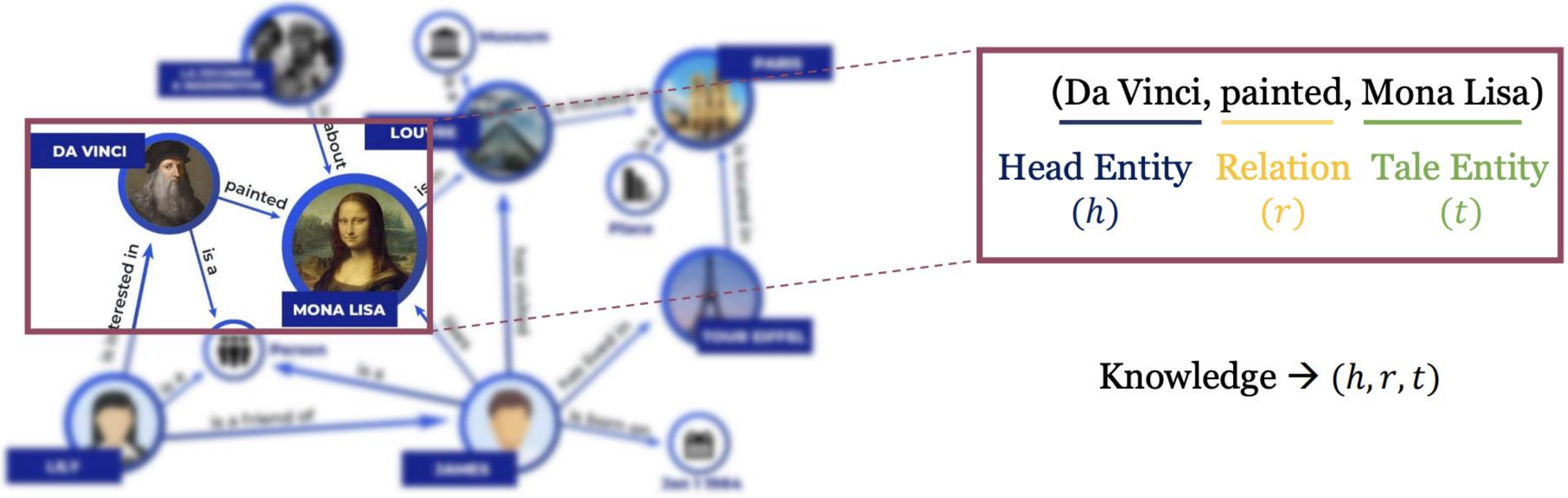


Knowledge Graph

Knowledge enhanced NLP

#02 Techniques to add knowledge to LMs

Asides: Knowledge Graph (KG)



Knowledge Graph

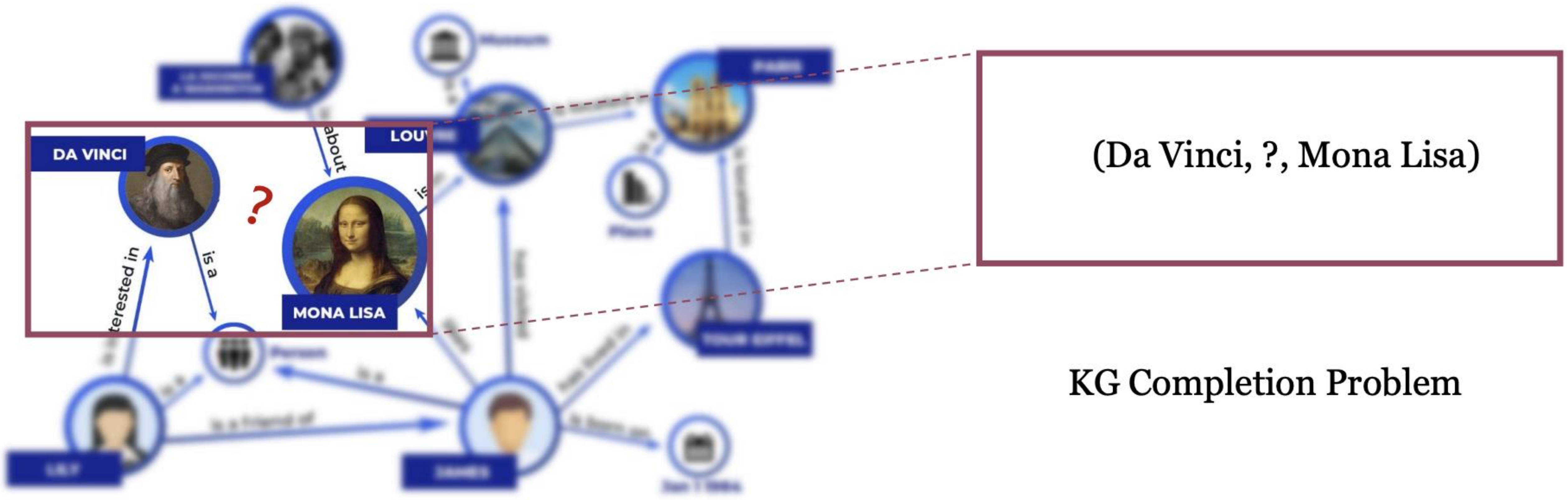
Knowledge → (h, r, t)

Knowledge enhanced NLP

#02 Techniques to add knowledge to LMs

Asides: Knowledge Graph Embedding

- Knowledge Graph Embedding
  - Knowledge Graph Completion 문제 해결을 위한 방법
  - 모든 Entity와 Relation을 낮은 차원 벡터로 표현, 벡터 연산을 통해 Graph Completion 수행
  - Knowledge Enhanced LMs에 직접적으로 사용됨



Knowledge Graph

KG Completion Problem

Knowledge enhanced NLP

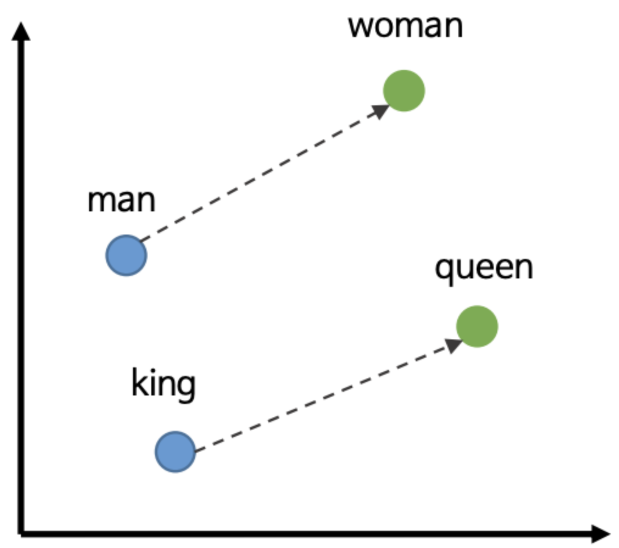
#02 Techniques to add knowledge to LMs

TransE (Translating Embeddings for Modeling Multi-relational Data)

TransE(2013)

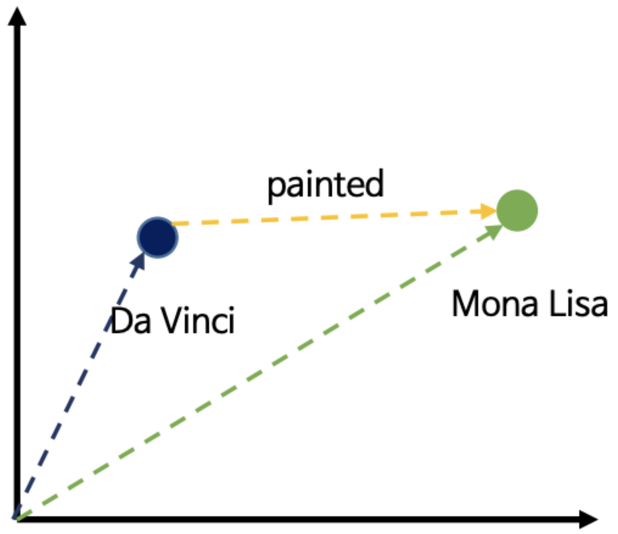
- 대표적인 Knowledge Graph Embedding 방법론
- 참 $h, r, t \rightarrow h + r = t$ 가 되도록 유도
- 거짓 $(h, r, t) \rightarrow h + r \neq t$ 가 되도록 유도

Word Embedding
(Word2Vec, Glove 등)



- 의미구조 유지
- Co-occurrence 기반

KG Embedding
(TransE)



- 논리구조 유지
- 지식 기반

Knowledge enhanced NLP

#02 Techniques to add knowledge to LMs

Aside: TransE (Translating Embeddings for Modeling Multi-relational Data)

1. Introduction

해당 논문은 지식 그래프의 벡터 임베딩의 포문을 연 글이다. 이 글 이후에 TransH, TransR 등 후속 논문들이 등장했고 지식그래프와 Question Answering 등등 여러 도메인에서 기초가 되었다고도 볼 수 있다.

1-1. Knowledge Graph

먼저 지식 그래프를 간단하게 살펴보도록 하자. 지식 그래프는 `Multi-relational data` 라고 볼 수 있으며 기본적으로 `entity` 와 그 사이의 `relation` 으로 구성되어 있다.

간단히 말해 그래프의 일종이라고 생각하면 된다! 또한 지식 그래프에서 사용하는 데이터 표현 방법을 알아 두고 가는 것이 중요하다!

(h, l, t) -> head, label, tail

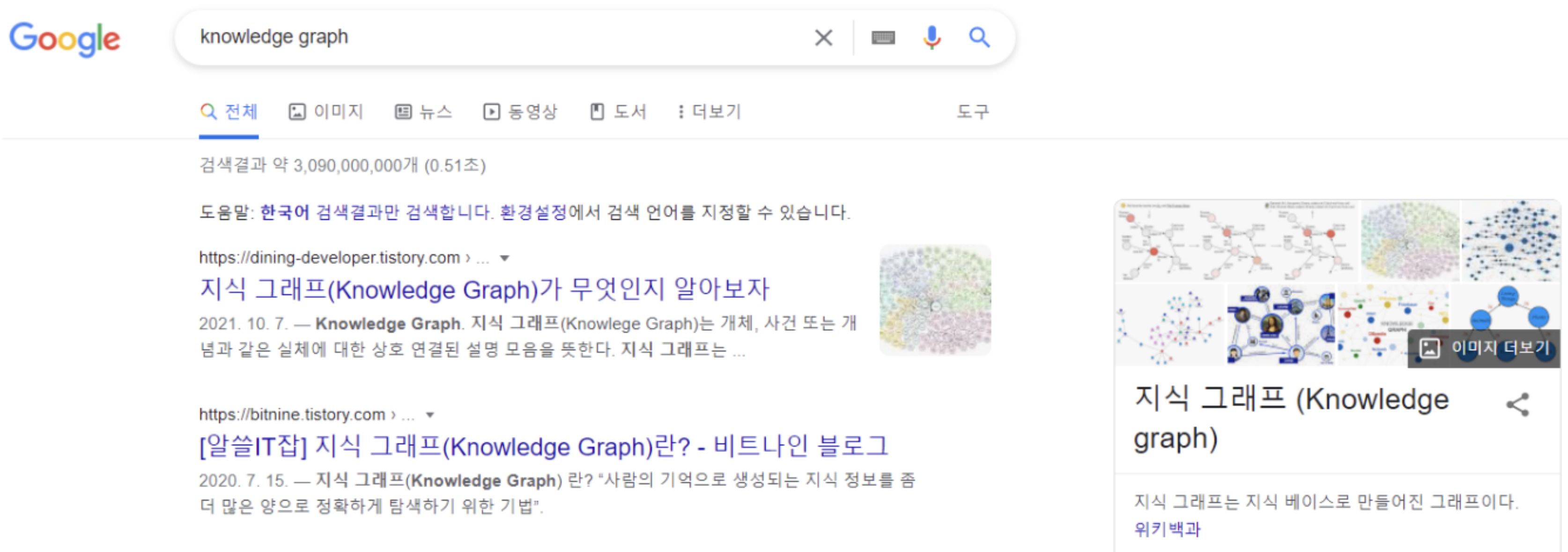
head와 tail은 두 엔티티이며, label은 두 엔티티 간의 관계이다. 기본적으로 지식 그래프는 방향성을 가진 그래프로 정의된다.

head, label, tail의 예시를 하나만 들자면 다음과 같다.

> Seoul **(HEAD)** is capital of **(LABEL)** South Korea **(TAIL)**.

1-2. Modeling multi-relational data

이런 지식 그래프, 혹은 엔티티 간의 연결구조와 그 예측에 관한 시도는 여러 번 있어왔다. 이는 추천 시스템과도 연관이 있는데, 간단한 예로 구글 검색을 할 때 오른쪽에 뜨는 작은 창은 지식 그래프의 유명한 예시이다.



1-3 What is Translation?

`translation` 은 **TransE**의 핵심 요소이다. 그렇다면 과연 translation은 무엇일까? 저자는 head와 tail 간의 relationship vector를 translation vector라고 한다. 직관적으로 말하자면 ***어떠한 개체를 translation을 통해 다른 개체로 매핑한다*** 고 볼 수 있겠다.

TransE논문리뷰

#02 Techniques to add knowledge to LMs

Aside: TransE (Translating Embeddings for Modeling Multi-relational Data)

2. Translation-based model

2-1. Algorithm

TransE 모델의 알고리즘은 다음과 같다.

Initialization

1. 각각의 l (label) 에 대해 $Unif(-\frac{6}{\sqrt{k}}, \frac{6}{\sqrt{k}})$ 로 초기화한다.
2. 각각의 l 에 대해 vector normalizing을 시행한다.
3. 각각의 e (entity) 에 대해 $Unif(-\frac{6}{\sqrt{k}}, \frac{6}{\sqrt{k}})$ 로 초기화한다.

loop

1. 각각의 e (entity) 에 대해 vector normalizing을 시행한다.
2. Size b의 미니배치를 S (traning set) 에서 샘플링한다. -> S_{batch}
3. $T_{batch} = \emptyset$으로 초기화한다.
4. for문: corrupted triplet에서 S'를 샘플링하고, T_{batch}에 uncorrupted triplet과 합한다.
5. 임베딩을 업데이트한다.

$\sum_{((h,l,t),(h',l',t'))\in T_{batch}} \nabla[\gamma + d(h+l,t) - d(h'+l,t')]_+$

모델의 기본적인 가정은 다음과 같다:

> 1. head vector에 relation vector를 더하면 tail vector에 근접할 것이다!
> 2. 라벨 (h, l, t)에서 t는 $h + l$과 가장 가까운 이웃이어야 한다.

다시 말해 (이상적으로는) translation을 통해 head vector를 tail vector로 매핑할 수 있다는 뜻이 된다. 그렇다면 $(head + label - tail)$ 수식으로 distance function을 만들 수 있을 것이다. $l1$과 $l2$ measure를 모두 사용할 수 있다. euclidian distance를 사용하면 다음과 같이 표현할 수 있다:

$f = ||h + l - t||_2^2$

그리고 목적함수(objective function)으로는 margin-based ranking criterion을 사용하였다. 식은 다음과 같다:

$\sum_{(h,l,t)\in S} \sum_{(h',l',t')\in S'} \nabla[\gamma + d(h+l,t) - d(h'+l,t')]_+$

2-2 Corrupted Set

지식 그래프는 non-corrupted set만 저장이 되어 있다. 다시 말해 정답인 데이터만 가지고 있기 때문에, 원래 데이터로만은 학습이 불가능하다.

따라서 저자는 원래 데이터의 $head$나 $tail$을 랜덤하게 바꾸어서 corrupted set을 만들고, 여기서 샘플링을 하여 학습을 진행했다.

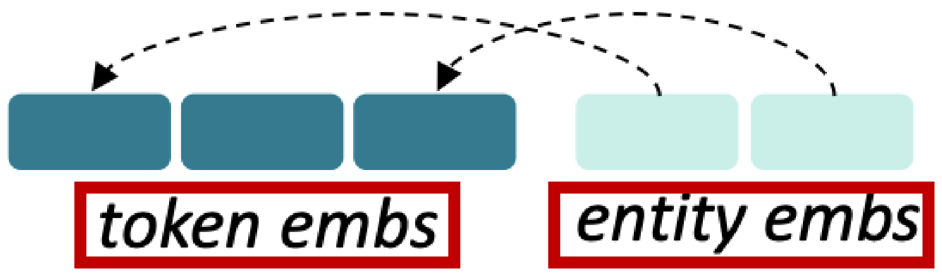
3. Conclusion

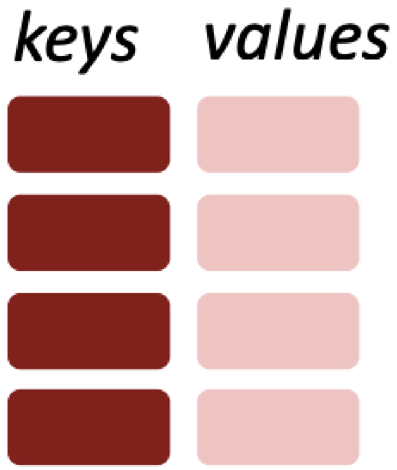
이렇게 지식 그래프의 벡터 임베딩 방법을 소개한 논문 TransE를 알아보았다. 다음 포스트에서는 TransE의 한계와 후속 논문들을 살펴볼 것이다.

TransE논문리뷰

#02 Techniques to add knowledge to LMs

Techniques to add knowledge to LMs





1. Add pretrained entity embeddings
- ERNIE

2. Use an external memory
- KGLM
- kNN-LM

3. Modify the training data
- WKLM
- ERNIE (another!), salient span masking

Cs224n_slides

#02 Techniques to add knowledge to LMs

1-1. ERNIE: Model Architecture

Question: How do we incorporate pretrained entity embeddings from a different embedding space?
Answer: Learn a fusion layer to combine context and entity information.

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

We assume there's a known alignment between entities and words in the sentence such that $e_k = f\left(w_j\right)$

- $\boldsymbol{w}_j$ is the embedding of word j in a sequence of words
- $\boldsymbol{e}_k$ is the corresponding entity embedding

- Text representation과 entity representation은 서로 다른 embedding space로부터 생성됨
- ERNIE에서 text는 BERT word embedding으로, entity는 pretrained entity embedding으로 추출 (TransE)
- 서로 다른 embedding space를 united feature space로 결합하기 위해 추가적인 fusion layer 사용

#02 Techniques to add knowledge to LMs

1-1. ERNIE: Model Architecture

- Text encoder:

  multi-layer bidirectional Transformer encoder over the words in the sentence

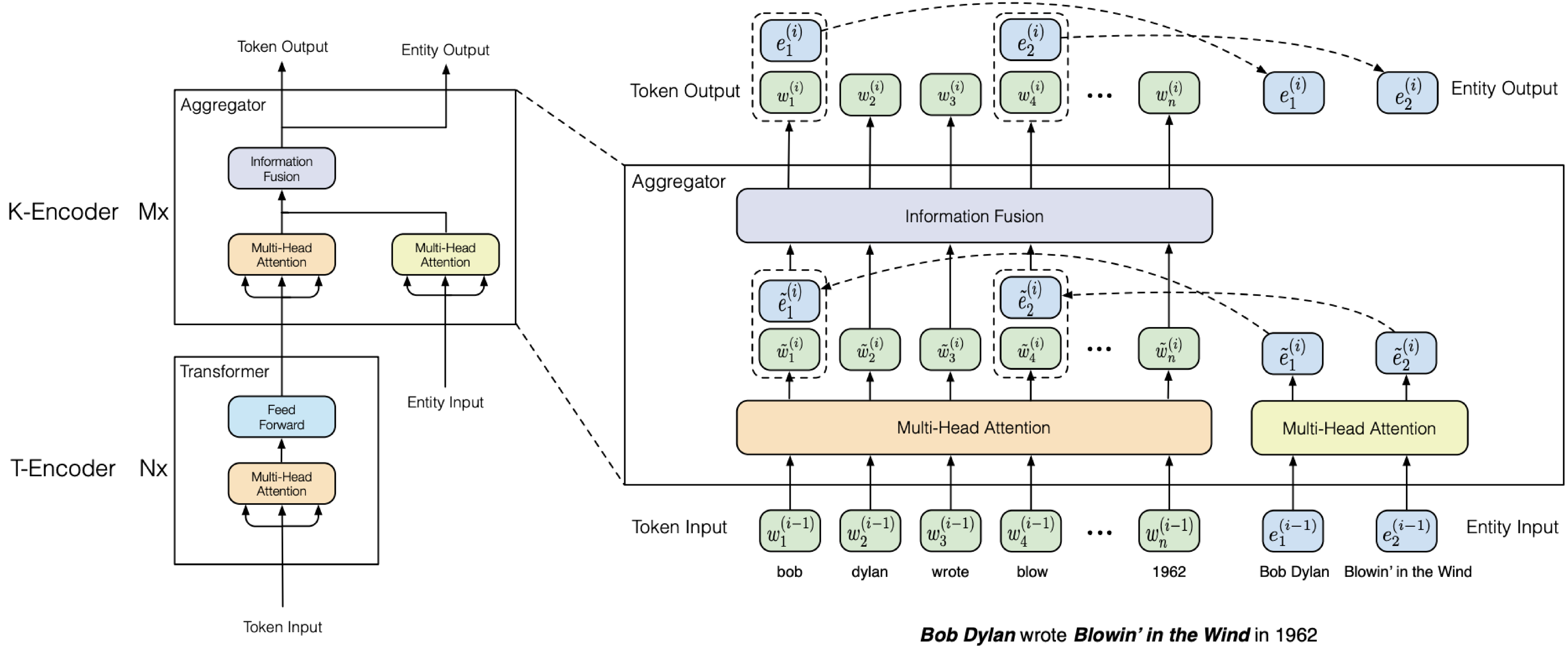
- Knowledge encoder: stacked blocks composed of:
  - Two multi-headed attentions (MHAs) over entity embeddings and token embeddings
  - A fusion layer to combine the output of the MHAs

Cs224n_slides

EWHA EURON

#02 Techniques to add knowledge to LMs

1-1. ERNIE: Enhanced Language Representation with Informative Entities



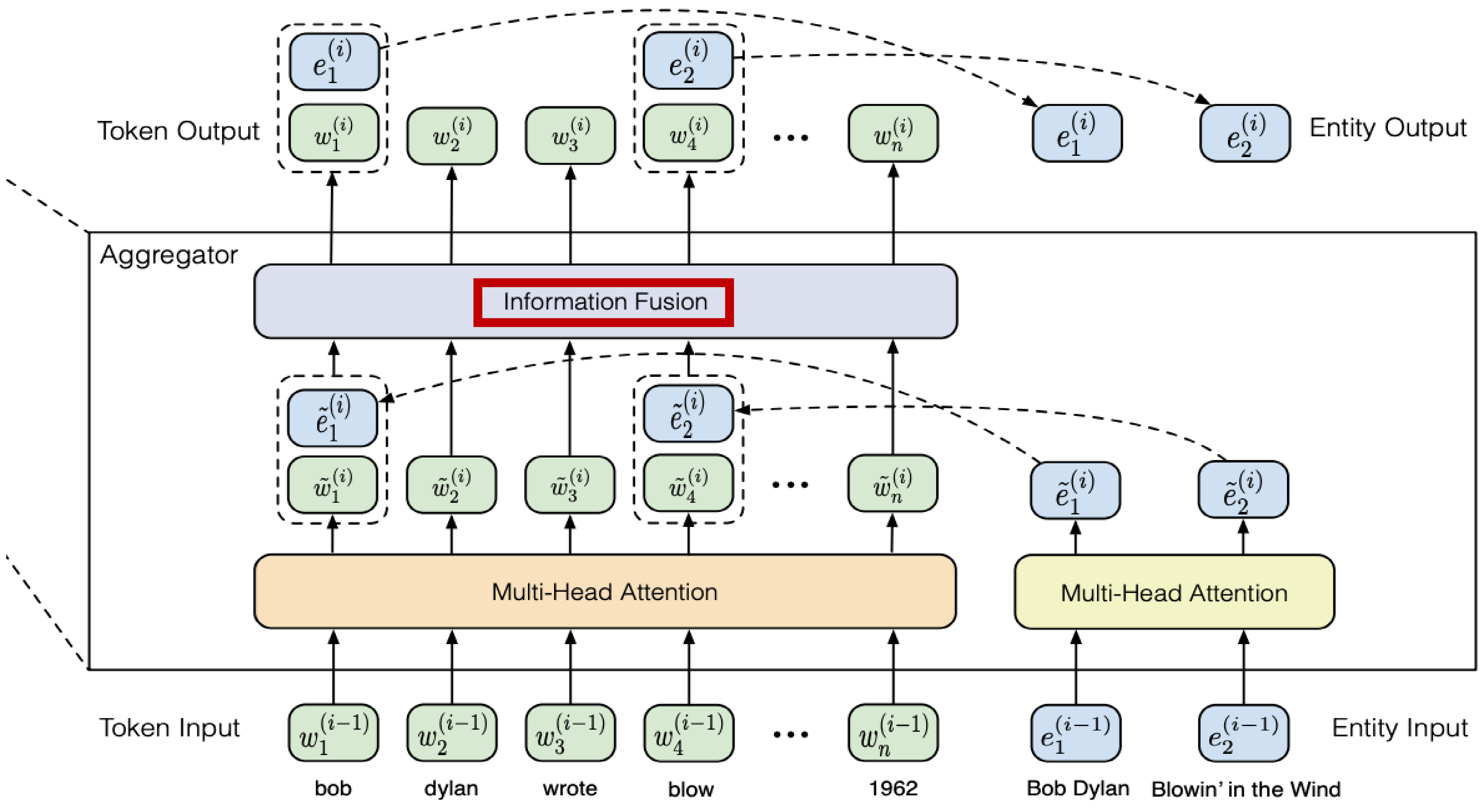
(a) Model Achitecture

(b) Aggregator

ERNIE

#02 Techniques to add knowledge to LMs

1-1. ERNIE: Enhanced Language Representation with Informative Entities



Bob Dylan wrote ***Blowin' in the Wind*** in 1962

(b) Aggregator

Information fusion process

$$
\begin{aligned}
\boldsymbol{h}_j &= \sigma(\tilde{\boldsymbol{W}}_t^{(i)}\tilde{\boldsymbol{w}}_j^{(i)} + \tilde{\boldsymbol{W}}_e^{(i)}\tilde{\boldsymbol{e}}_k^{(i)} + \tilde{\boldsymbol{b}}^{(i)}),\\
\boldsymbol{w}_j^{(i)} &= \sigma(\boldsymbol{W}_t^{(i)}\boldsymbol{h}_j + \boldsymbol{b}_t^{(i)}),\\
\boldsymbol{e}_k^{(i)} &= \sigma(\boldsymbol{W}_e^{(i)}\boldsymbol{h}_j + \boldsymbol{b}_e^{(i)}).
\end{aligned} \tag{4}
$$

$$
\begin{aligned}
\boldsymbol{h}_j &= \sigma(\tilde{\boldsymbol{W}}_t^{(i)}\tilde{\boldsymbol{w}}_j^{(i)} + \tilde{\boldsymbol{b}}^{(i)}),\\
\boldsymbol{w}_j^{(i)} &= \sigma(\boldsymbol{W}_t^{(i)}\boldsymbol{h}_j + \boldsymbol{b}_t^{(i)}).
\end{aligned} \tag{5}
$$

Aggregation operation

$$
\begin{aligned}
\{\boldsymbol{w}_1^{(i)}, \ldots, \boldsymbol{w}_n^{(i)}\}, \{\boldsymbol{e}_1^{(i)}, \ldots, \boldsymbol{e}_m^{(i)}\} = \texttt{Aggregator}(&\\
\{\boldsymbol{w}_1^{(i-1)}, \ldots, \boldsymbol{w}_n^{(i-1)}\}, \{\boldsymbol{e}_1^{(i-1)}, \ldots, \boldsymbol{e}_m^{(i-1)}\}).&
\end{aligned} \tag{6}
$$

ERNIE

#02 Techniques to add knowledge to LMs

1-1. ERNIE: Enhanced Language Representation with Informative Entities

For incorporating external knowledge into language representation models, there are two main challenges.

(1) Structured Knowledge Encoding

regarding to the given text, how to effectively extract and encode its related informative facts in KGs for language representation models is an important problem;

텍스트 안의 Named entity mentions와 Knowledge Graph의 entity를 align해서 정보를 얻음
TransE(knowledge graph embedding 방법론)를 사용하여 entity를 위한 embedding 진행
Entity representation을 기존 모델에 적절히 통합시킴

(2) Heterogeneous Information Fusion

the pre-training procedure for language representation is quite different from the knowledge representation procedure, leading to two individual vector spaces. How to design a special pre-training objective to fuse lexical, syntactic, and knowledge information is another challenge.

Entity representation을 고려하기 위한 새로운 object를 제안하여 각각의 vector space를 적절히 fusion함
Entity representation + token representation = knowledgeable language representation

ERNIE

#02 Techniques to add knowledge to LMs

1-1. ERNIE: Enhanced Language Representation with Informative Entities

- Pretrain with three tasks:
  - Masked language model and next sentence prediction (i.e., BERT tasks)
  - Knowledge pretraining task (dEA1): randomly mask token-entity alignments and predict corresponding entity for a token from the entities in the sequence

$$p(e_j \mid w_i) = \frac{\exp(\boldsymbol{W}\boldsymbol{w}_i \cdot \boldsymbol{e}_j)}{\sum_{k=1}^{m} \exp(\boldsymbol{W}\boldsymbol{w}_i \cdot \boldsymbol{e}_k)}$$

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

ERNIE

#02 Techniques to add knowledge to LMs

1-1. ERNIE: Enhanced Language Representation with Informative Entities

Strengths:

- Combines entity + context info through fusion layers and a knowledge pretraining task
- Improves performance downstream on knowledge-driven tasks

Remaining challenges:

- Needs text data with entities annotated as input, even for downstream tasks
  - For instance, “Bob Dylan wrote Blowin’ in the Wind” needs entities pre-linked to input entities into ERNIE
- Requires further (expensive) pretraining of the LM1

Cs224n_slides

#02 Techniques to add knowledge to LMs

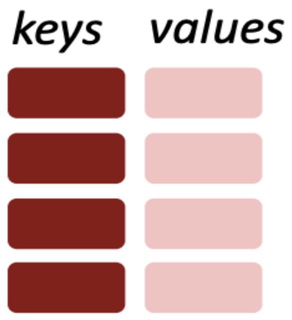
2. Use an external memory

- Previous methods rely on the pretrained entity embeddings to encode the factual knowledge from KBs for the language model.

- Question: Are there more direct ways than pretrained entity embeddings to provide the model factual knowledge?

- Answer: Yes! Give the model access to an external memory (a key-value store with access to KG triples or context information)

- Advantages:
  - Can better support injecting and updating factual knowledge
    - Often without more pretraining!
  - More interpretable



Use an external memory

- KGLM
- kNN-LM

Cs224n_slides

#02 Techniques to add knowledge to LMs

2-1. Using Knowledge-Graphs for Fact-Aware Language Modeling (KGLM)

- Key idea: condition the language model on a knowledge graph (KG)
- Recall that language models predict the next word by computing

$$\mathrm{P}\left(x^{(t+1)} \middle| x^{(t)}, \ldots, x^{(1)}\right), \text{ where } x^{(1)}, \ldots, x^{(t)} \text{ is a sequence of words}$$

- Now, predict the next word using entity information, by computing

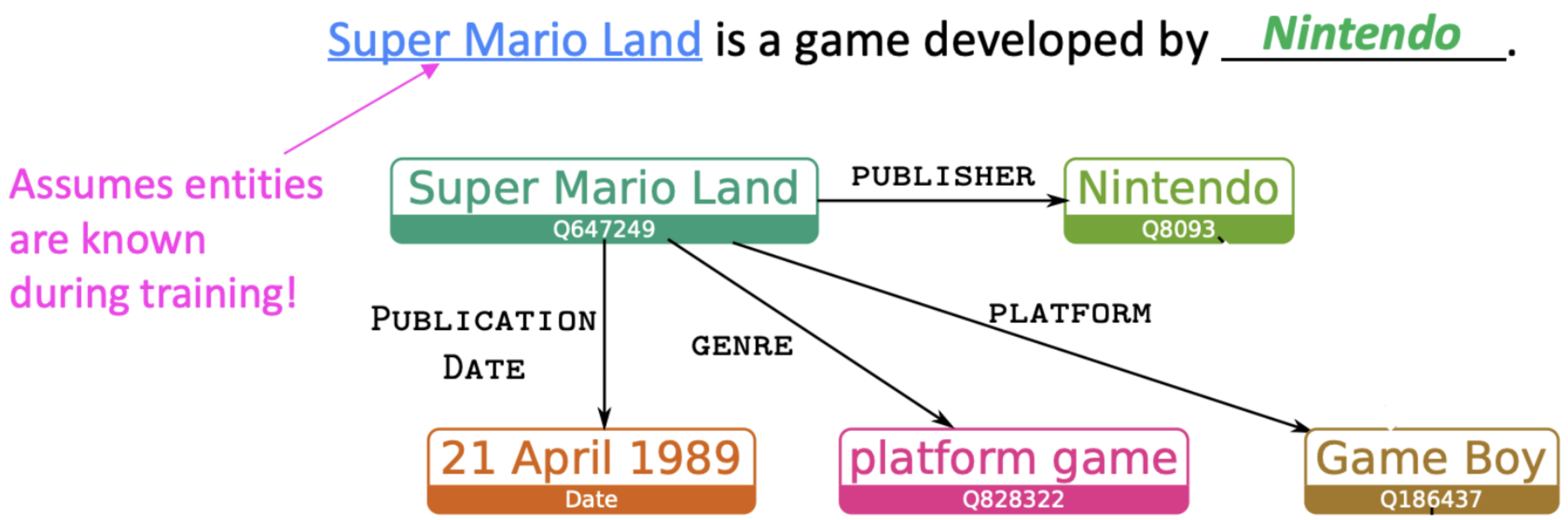
$$\mathrm{P}\left(x^{(t+1)}, \mathcal{E}^{(t+1)} \middle| x^{(t)}, \ldots, x^{(1)}, \mathcal{E}^{(t)}, \ldots, \mathcal{E}^{(1)}\right)$$

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

#02 Techniques to add knowledge to LMs

2-1. KGLM

- Build a "local" knowledge graph as you iterate over the sequence

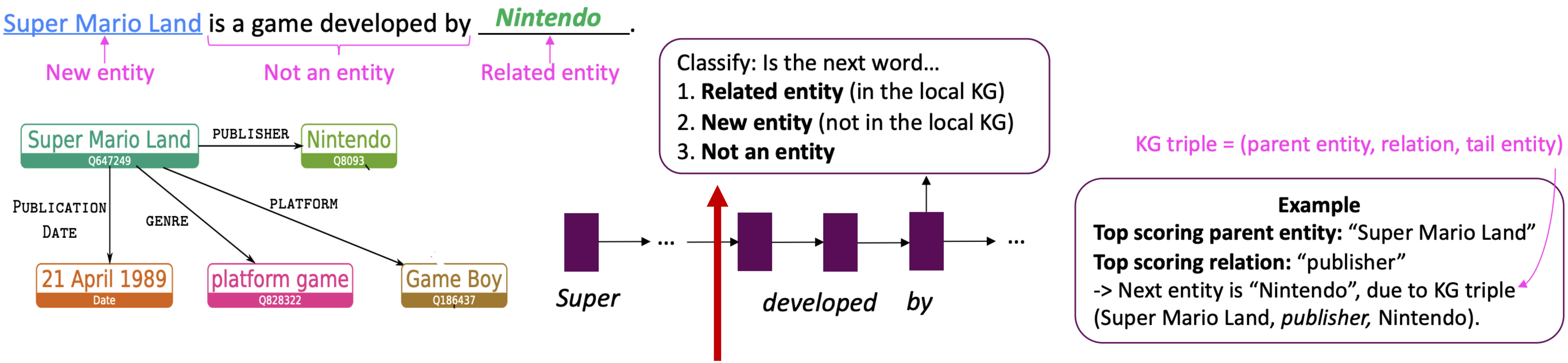


- When should the LM use the local KG to predict the next word?

Using Knowledge Graphs for Fact-Aware Language Modeling

#02 Techniques to add knowledge to LMs

2-1. KGLM



- Use the LSTM hidden state to predict the type of the next word (3 classes)
- **How** does the LM predict the next entity and word in each case?

- KGLM은 Fact Completion Task에서 GPT-2, AWD-LSTM을 넘는 성능을 보임
- 특히 GPT-2는 일반적인 token을 내뱉는 반면에, KGLM은 좀 더 정확하고 세밀한 token을 내뱉음, 또한 정보를 수정할 수 있다는 장점 있음

Cs224n_slides

#02 Techniques to add knowledge to LMs

2-2. More recent takes: k-Nearest Neighbor Language Models (kNN-LM)

- 다음 단어를 예측하는 것보다, text sequences 사이의 유사도를 학습하는 것이 더 쉽다 라는 생각에서 출발함

- So, store all representations of text sequences in a nearest neighbor datastore!

- At inference:
  1. Find the k most similar sequences of text in the datastore
  2. Retrieve the corresponding values (i.e. the next word) for the k sequences
  3. Combine the kNN probabilities and LM probabilities for the final prediction

$$P(y|x) = \lambda P_{kNN}(y|x) + (1-\lambda)P_{LM}(y|x)$$

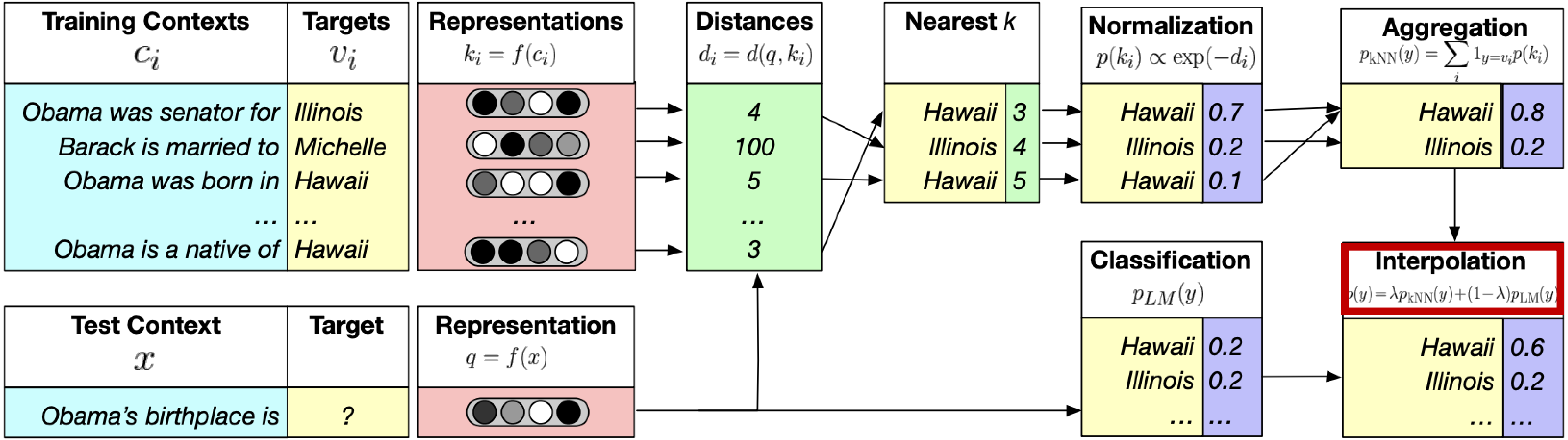
Cs224n_slides

#02 Techniques to add knowledge to LMs

2-2. More recent takes: k-Nearest Neighbor Language Models (kNN-LM)

Example: Shakespeare's play ______ ….
Task: Predict the next word with kNN-LM

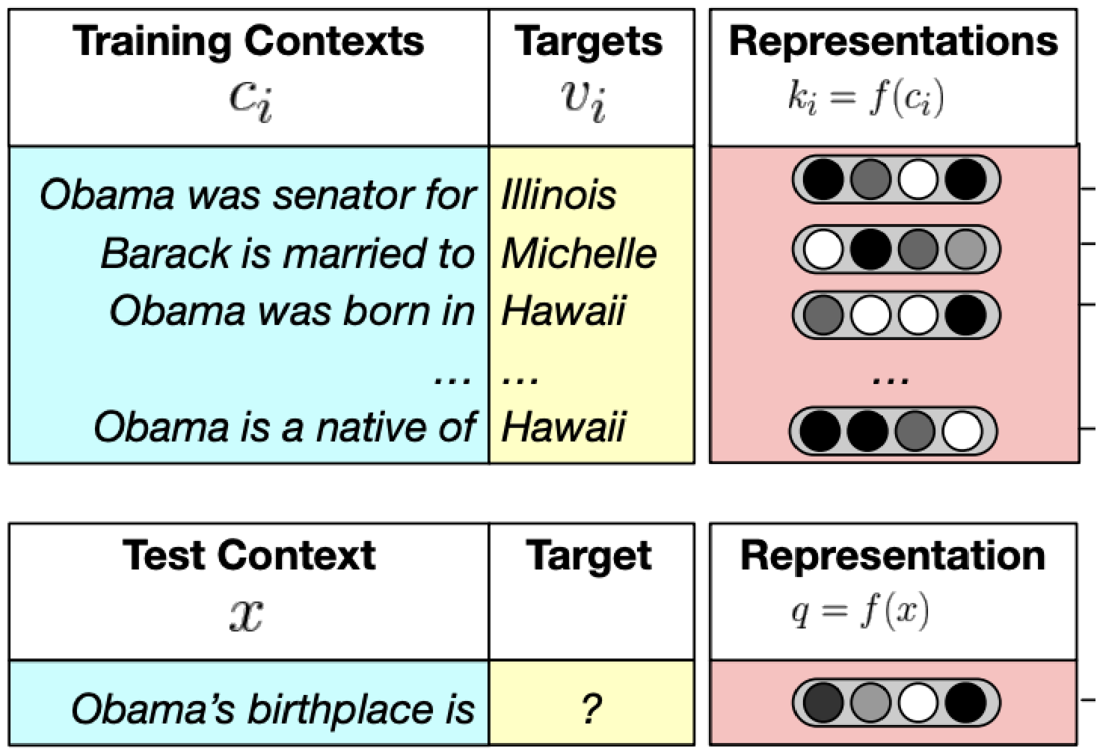
1. Find the k most similar sequences of text in the **datastore**
2. Retrieve the corresponding values (i.e. the next word) for the k sequences
3. Combine the **kNN probabilities and LM probabilities** for the final prediction



Cs224n_slides

#02 Techniques to add knowledge to LMs

2-2. More recent takes: k-Nearest Neighbor Language Models (kNN-LM)



Datastore Let $f(\cdot)$ be the function that maps a context c to a fixed-length vector representation computed by the pre-trained LM. For instance, in a Transformer LM, $f(c)$ could map c to an intermediate representation that is output by an arbitrary self-attention layer. Then, given the i-th training example $(c_i, w_i) \in \mathcal{D}$, we define the key-value pair (k_i, v_i), where the key k_i is the vector representation of the context $f(c_i)$ and the value v_i is the target word w_i. The datastore $(\mathcal{K}, \mathcal{V})$ is thus the set of all key-value pairs constructed from all the training examples in $\mathcal{D}$:

$$(\mathcal{K}, \mathcal{V}) = \{(f(c_i), w_i) | (c_i, w_i) \in \mathcal{D}\} \tag{1}$$

#02 Techniques to add knowledge to LMs

#3 Modify the training data

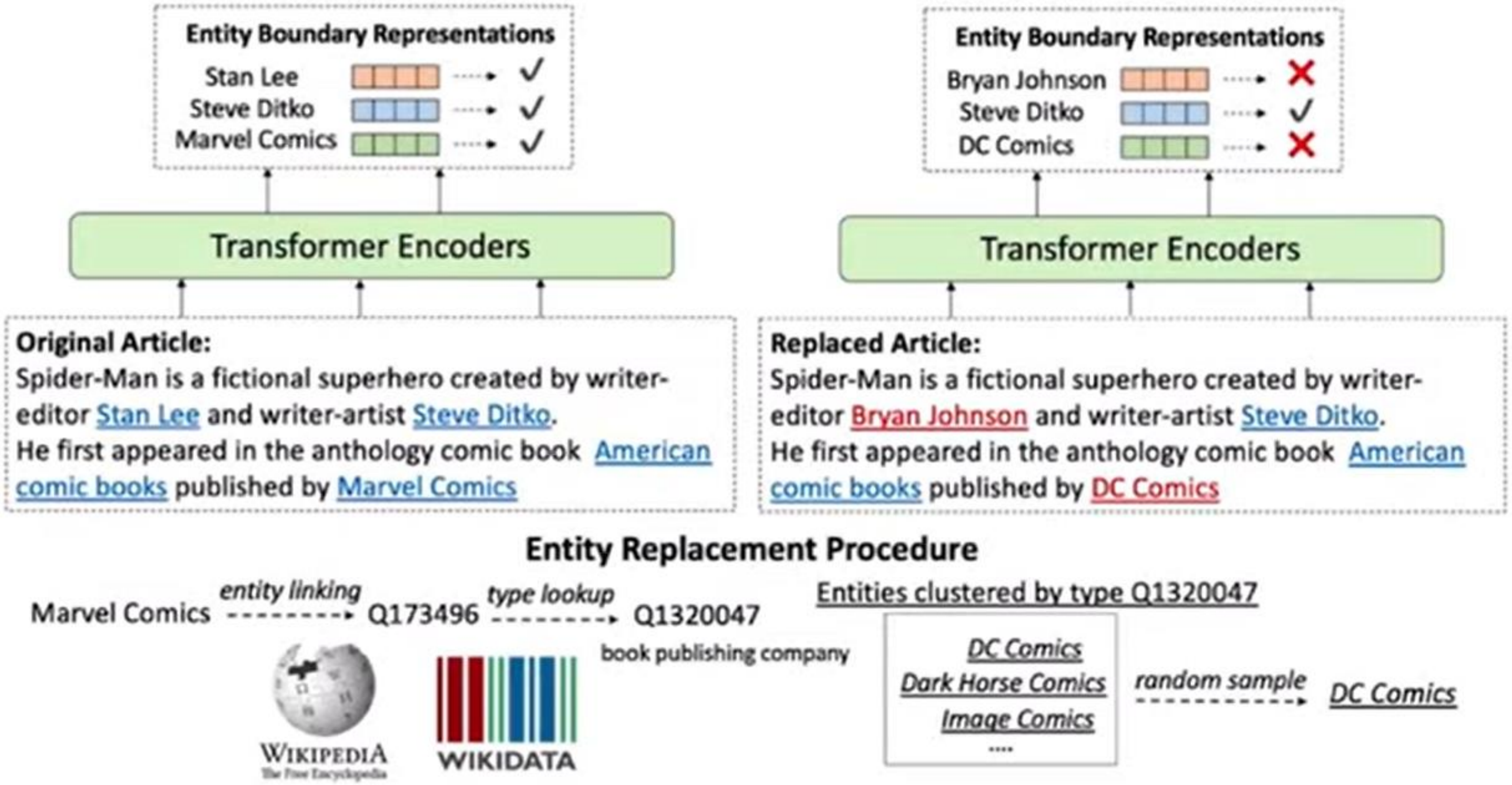
- Pretraining 과정에서 자연스럽게 knowledge를 주입
- 직접 knowledge를 infuse/inject 하기 보다 학습과정에서 자연스럽게 knowledge를 배울 수 있다.

1. WKLM : Weakly Supervised Knowledge Pretrained Language Model
   - 모델이 true knowledge와 false knowledge를 구분할 수 있도록 학습
   - 도출된 값이 true인지 학습하기 위해 기존 데이터를 사용하여 false knowledge를 생성
     1) 특정 entity와 동일한 type의 entity를 활용하여 기존 문장을 변경
     2) 새롭게 만들어진 문장을 negative knowledge statement라 한다

#02 Techniques to add knowledge to LMs

#3 Modify the training data

1. WKLM : Weakly Supervised Knowledge Pretrained Language Model



#02 Techniques to add knowledge to LMs

#3 Modify the training data

1. WKLM : Weakly Supervised Knowledge Pretrained Language Model

- Entity replacement loss

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

- Total loss

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

#02 Techniques to add knowledge to LMs

#3 Modify the training data

2. ERNIE : Enhanced Representation through Knowledge Integration

- 바이두에서 발표한 ERNIE
- 최근 3.0 모델을 발표하며 지속적으로 발전하는 중

- Knowledge masking strategy로 정보 주입
- 중국어에 대해서만 실험

- 여러가지 task를 학습하는 continual multitask training framework 제안
- 영어 모델도 발표함

- Universal representation
- Task specific representation
- Long sequence text를 다룸

#02 Techniques to add knowledge to LMs

#3 Modify the training data

2. ERNIE : Enhanced Representation through Knowledge Integration

칭화대 ERNIE와 바이두 ERNIE의 차이점

- 칭화대 ERNIE : entity 정보를 주입하기 위해 pretrained entity embedding을 사용함
- 바이두 ERNIE : entity 정보를 주입하기 위해 별도의 entity embedding을 사용하지 않고 masking 방법을 통해 information 주입

EWHA EURON

#02 Techniques to add knowledge to LMs

#3 Modify the training data

3. REALM : Retrieval-Augmented Language Model Pre-Training

- Retriever와 reader를 한번에 학습하는 것을 제안
- Retriever를 Pretraining에서 수행하는 모델

[Main Contribution]

- Retriever와 reader를 한번에 학습하는 end-to-end 모델
- Input을 넣어 output을 찾는 과정을 두 단계로 나누어 진행

1) Neural Knowledge Retriever

query → query 의 답이 될 수 있는 document를 찾음

2) Knowledge-Augmented Encoder

Retrieved Document → Answer

#02 Techniques to add knowledge to LMs

#3 Modify the training data

3. REALM : Retrieval-Augmented Language Model Pre-Training

[Pretrained LM의 능력과 한계]

- Pretrained LM은 pretrain 단계에서 이미 large corpora로 학습되므로 <u>대량의 정보를 포함</u>
- 대부분의 Pretrained LM은 Cloze task로 학습을 진행하기 때문에 mask를 예측하는 과정에서 <u>언어를 이해할 뿐만 아니라 정보까지 습득할 수 있음</u>

But, pretrained LM이 정보를 저장하는 방식은 implicitly 하다

- Network에 어떤 knowledge가 학습되어 있는지 알 수 없음
- 더 많은 knowledge를 학습하기 위해서는 <u>model size를 증가</u> 시켜야 하며, <u>계산 비용이 상당</u>하다

#02 Techniques to add knowledge to LMs

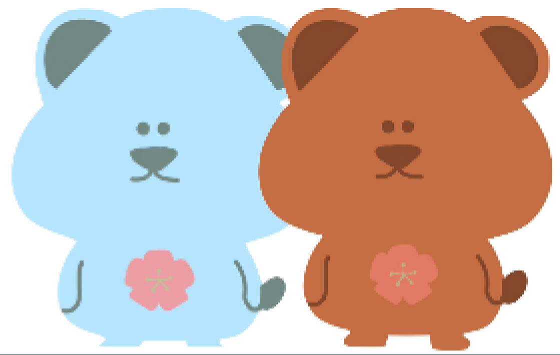
#3 Modify the training data

3. REALM : Retrieval-Augmented Language Model Pre-Training

Implicitly하기 때문에 explicit하게 knowledge를 학습 및 저장하는 모델이 필요함

- Textual knowledge retrieve를 통해 기존 pretrained LM 보다 해석 가능하고 explicit하게 knowledge를 학습하는 모델로 개선
- Retriever 과정이 pretraining에 포함되어 있는 형태이다
- 문장 → retriever → 정답을 찾아낼 수 있는 새로운 모델 구조

#03 Evaluating knowledge in LMs



#03 Evaluating knowledge in LMs

#1 LAMA(Language Model Analysis) probe

- 동일한 setting에서 학습하여 어떤 LM이 가장 많은 정보를 포함하는지 비교
- 하나의 benchmark로서 factual and commonsense knowledge를 probe함
- Unsupervised BERT가 factual knowledge를 학습한다고 주장

#2 LAMA-UHN

- LAMA에서 relational knowledge없이 답변 가능한 예제 모두 제거하고 새로운 데이터 생성
- UHN에서는 entity가 있어야 답변 가능한 데이터만 남아 있다.
- BERT가 entity name의 surface form에 지나치게 의존

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

프랑스 사람,
Italian-sounding name

#03 Evaluating knowledge in LMs

#3 prompt and performance

Prompt: input으로 주입되는 데이터의 형식

- 성능은 Prompt의 형식에 따라 많이 영향을 받는다
- LM은 input의 query에 따라 sensitive하다

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

THANK YOU