VL Deep Learning for Natural Language Processing

19. Enhancing Language Models

Prof. Dr. Ralf Krestel

AG Information Profiling and Retrieval

Recap: Language Models

- 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!

Recap: Language Models

- Traditionally, LMs are used for many tasks involving **generating** or **evaluating the probability** of text:
  - Summarization
  - Dialogue
  - Autocompletion
  - Machine translation
  - Fluency evaluation
  - ...
- 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**?

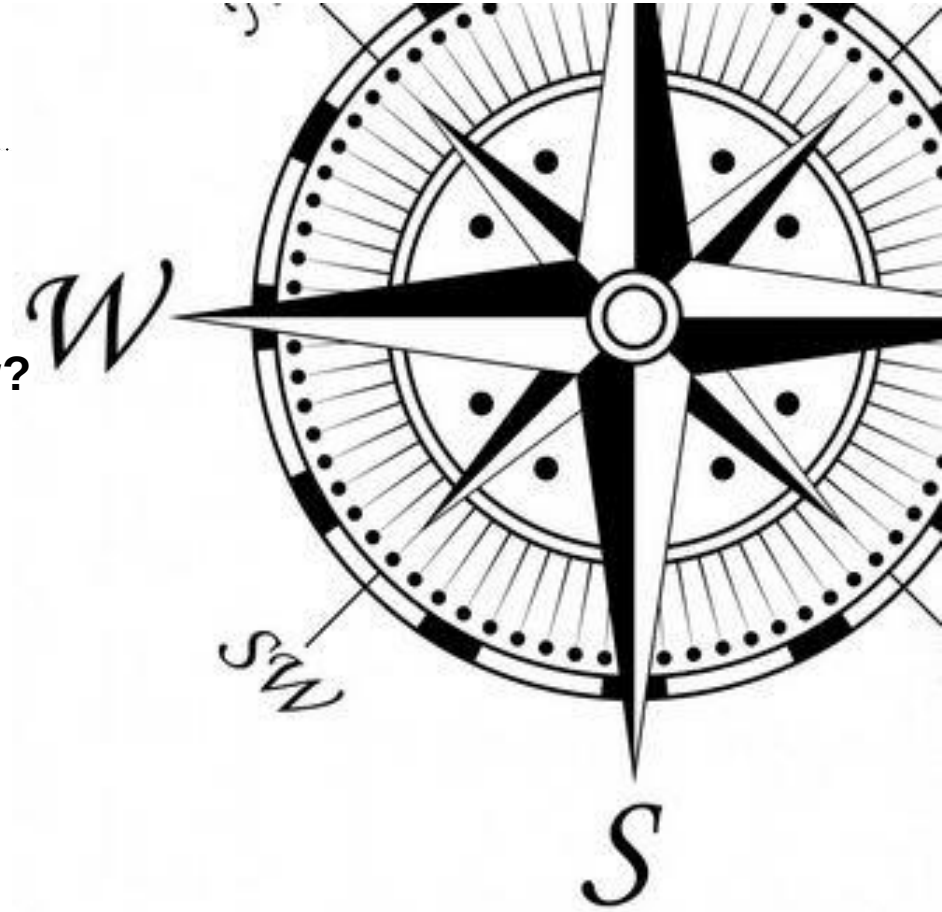
Learning Goals for this Chapter

- Understand how and what knowledge can be encoded in language models
- Know techniques to infuse knowledge into language models
- Be able to evaluate LMs for their contained knowledge

- Relevant Chapters:
  - S15 (2021): https://www.youtube.com/watch?v=y68RJVfGoto

Topics Today

1. **What Does a Language Model Know?**
2. Techniques to Add Knowledge to LMs
3. Evaluating Knowledge in LMs



What Does a Language Model Know?

- E.g. BERT-Large:
  - iPod Touch is produced by **Apple**.
  - London Jazz Festival is located in **London** .
  - Dani Alves plays with **Santos** .
  - Carl III used to communicate in **German** .
  - Ravens can **fly** .

Petroni, F., Rocktäschel, T., Riedel, S., Lewis, P., Bakhtin, A., Wu, Y., & Miller, A. (2019). Language Models as Knowledge Bases?. In *EMNLP-IJCNLP* (pp. 2463-2473).

What Does a Language Model Know?

- Takeaway: 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 is sensitive to the phrasing of **the prompt**
    - Correctly answers "x was *made* in y" templates but not "x was *created* in y"

- The inability to **reliably** recall knowledge is a key challenge facing LMs today!
  - Recent works have found LMs can recover **some** knowledge, but still far to go…

The Importance of Knowledge-Aware LMs

- 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
  - We'll come back to this when talking about evaluation!

- 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 LMs to have high quality on recalling facts

How to Store Knowledge (Traditionally)?

- In a **Database**
  - Relational / Graph / NoSQL DB
- In an **Ontology**
  - Metadata and how they relate (triples)
- In a **Knowledge Base**
  - Data and how they relate (fact triples)
- In a **Knowledge Graph**
  - Connecting the fact triples

Example Database

Books

Title	Author	Publisher	Year Published	Followed By
To Kill a Mockingbird	Harper Lee	J. B. Lippincott Company	1960	Go Set a Watchman
Go Set a Watchman	Harper Lee	HarperCollins, LLC; Heinemann	2015	
The Picture of Dorian Gray	Oscar Wilde	J. B. Lippincott & Co.	1890	
2001: A Space Odyssey	Arthur C. Clarke	New American Library, Hutchinson	1968	

Publishers

Name	City	Country
J. B. Lippincott & Company	Philadelphia	United States
HarperCollins, LLC	New York City	United States
Heinemann	Portsmouth	United States
New American Library	New York City	United States
Hutchinson	London	United Kingdom

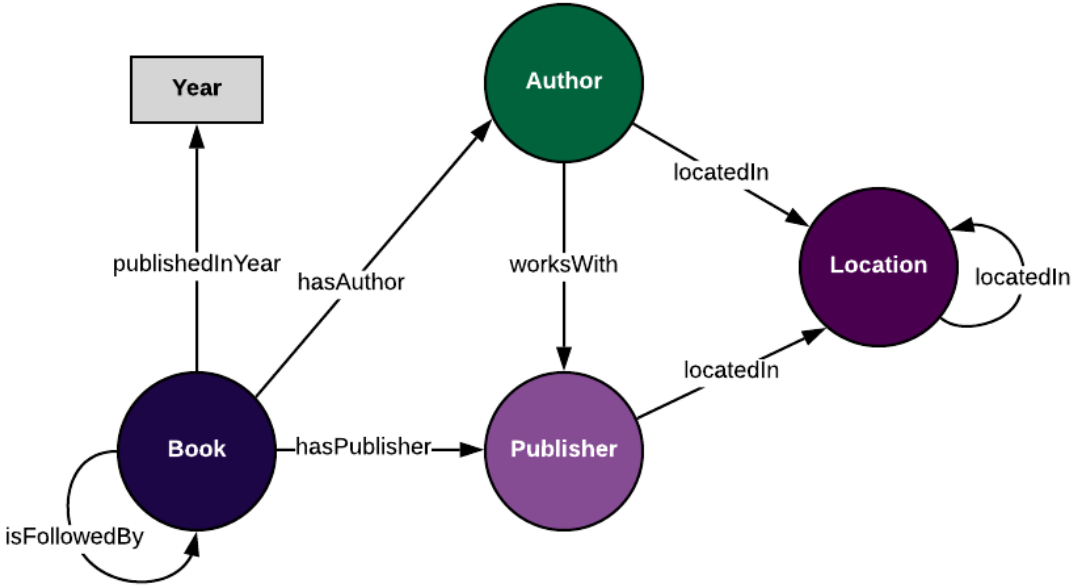
Authors

Name	Country of Birth
Harper Lee	United States
Oscar Wilde	Ireland
Arthur C. Clarke	United Kingdom

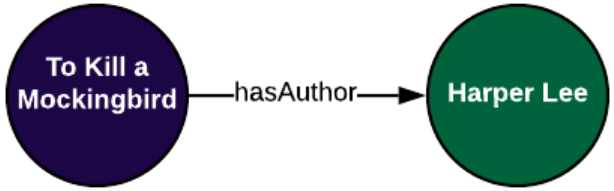
https://enterprise-knowledge.com/whats-the-difference-between-an-ontology-and-a-knowledge-graph/

Example Ontology

- Classes
  - Books
  - Authors
  - Publishers
  - Locations
- Attributes
  - Books are published on a date
  - …
- Relations
  - Books have authors
  - Books have publishers
  - Books are followed by sequels (other books)
  - …



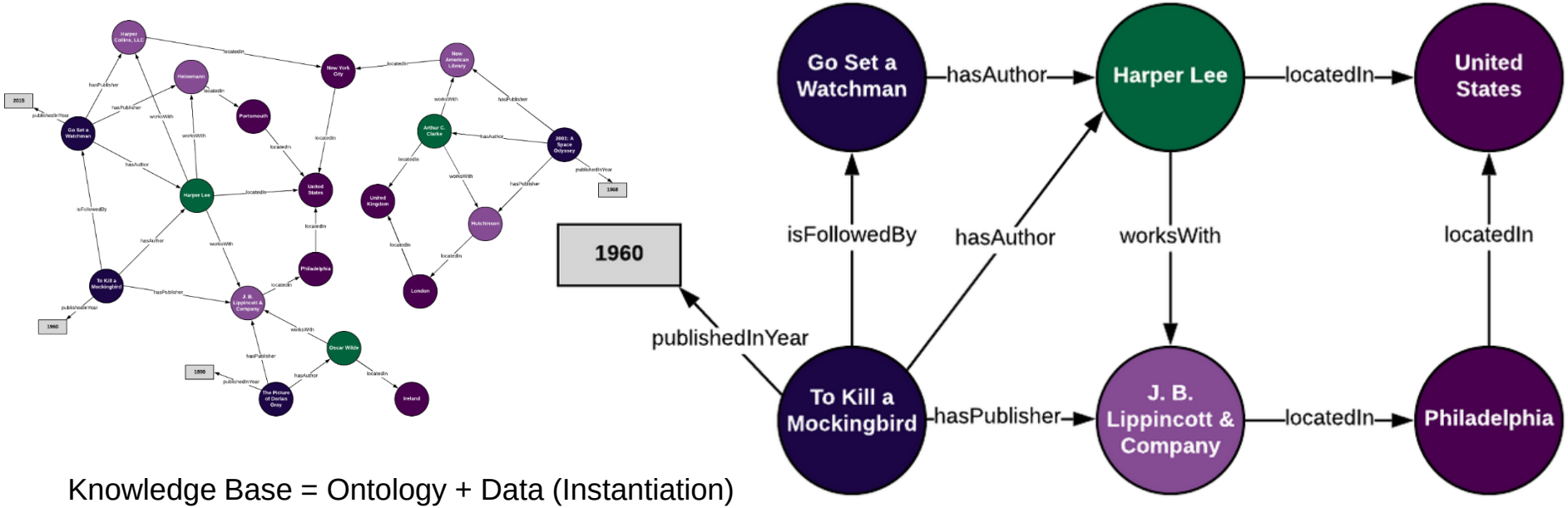
Example Knowledge Base



- Collection of facts:
  - To Kill a Mockingbird → has author → Harper Lee
  - To Kill a Mockingbird → has publisher→ JBL&C
  - To Kill a Mockingbird → published in → 1960
  - To Kill a Mockingbird → is followed by → Go Set a Watchman
  - Harper Lee → works with → JBL&C
  - JBL&C → located in → Philadelphia
  - Philadelphia → located in → United States of America
  - …

Example Knowledge Graph

- A graph containing all the triples of a knowledge base



Knowledge Base = Ontology + Data (Instantiation)

Knowledge Graph = KB + Graph

Knowledge Graph: Definition

- A Knowledge Graph is a data set that is:
  - **structured** (in the form of a specific data structure)
  - **normalized** (consisting of small units, such as vertices and edges)
  - **connected** (defined by the – possibly distant – connections between objects)

- Moreover, knowledge graphs are typically:
  - **explicit** (created purposefully with an intended meaning)
  - **declarative** (meaningful in itself, independent of a particular implementation or algorithm)
  - **annotated** (enriched with contextual information to record additional details and meta-data)
  - **non-hierarchical** (more than just a tree-structure)
  - **large** (millions rather than hundreds of elements)

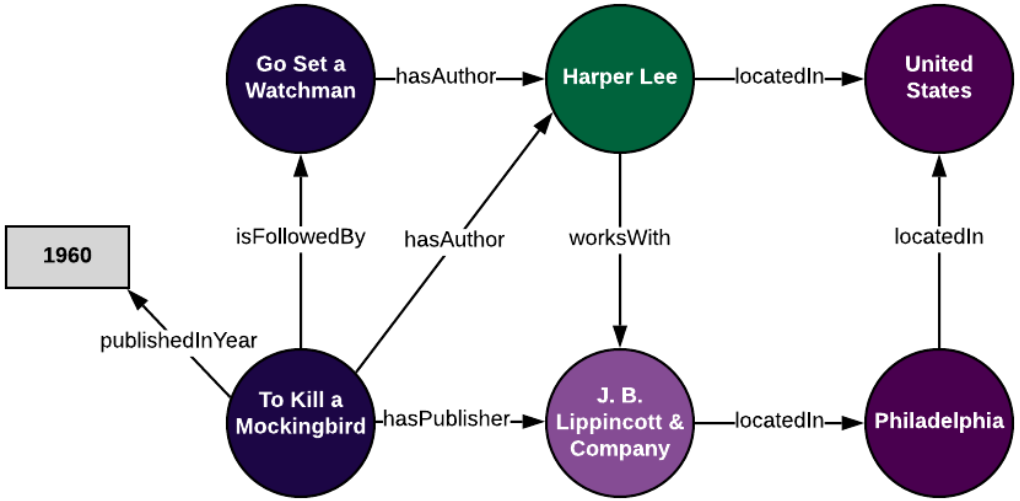
Markus Krötzsch https://iccl.inf.tu-dresden.de/w/images/b/bf/KG2018-Lecture-01-overlay.pdf

Knowledge Graph: (Counter-) Examples

- **Typical** knowledge graphs:
  - Wikidata, Yago, Freebase, DBpedia (though hardly annotated), OpenStreetMap
  - Google Knowledge Graph, Microsoft Bing Satori (presumably; we can't really know)
- **Debatable** cases:
  - Facebook's social graph: structured, normalized, connected, but not explicit (emerging from user interactions, without intended meaning beyond local relations)
  - WordNet: structured dictionary and thesaurus, but with important unstructured content (descriptions); explicit, declarative model
  - Global data from schema.org: maybe not very connected
  - Document stores (Lucene, MongoDB, …): structured, but not normalized; connections 2nd
- Primarily **not** knowledge graphs:
  - Wikipedia: mostly unstructured text; not normalized; connections (links) important but secondary (similar: the Web)
  - Relational database of company X: structured and possibly normalized, but no focus on connections (traditional RDBMS support connectivity queries only poorly)

Markus Krötzsch https://iccl.inf.tu-dresden.de/w/images/b/bf/KG2018-Lecture-01-overlay.pdf

Querying Traditional Knowledge Bases / Graphs

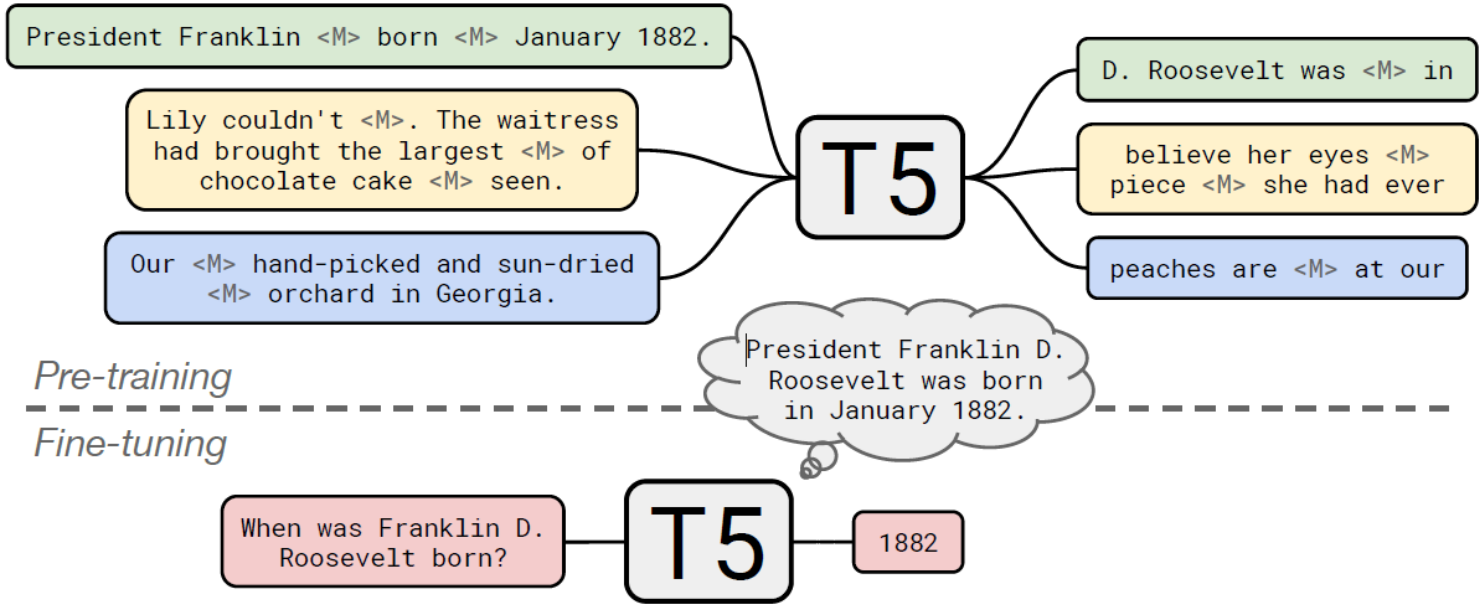


- Query knowledge base with SQL statements

```
SELECT publishedInYear
WHERE book = "To Kill a Mockingbird"
```

Querying LMs as Knowledge Bases

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



Roberts, A., Raffel, C., & Shazeer, N. (2020). How Much Knowledge Can You Pack Into the Parameters of a Language Model? In *EMNLP* (pp. 5418-5426).

Advantages of LMs over Traditional KBs

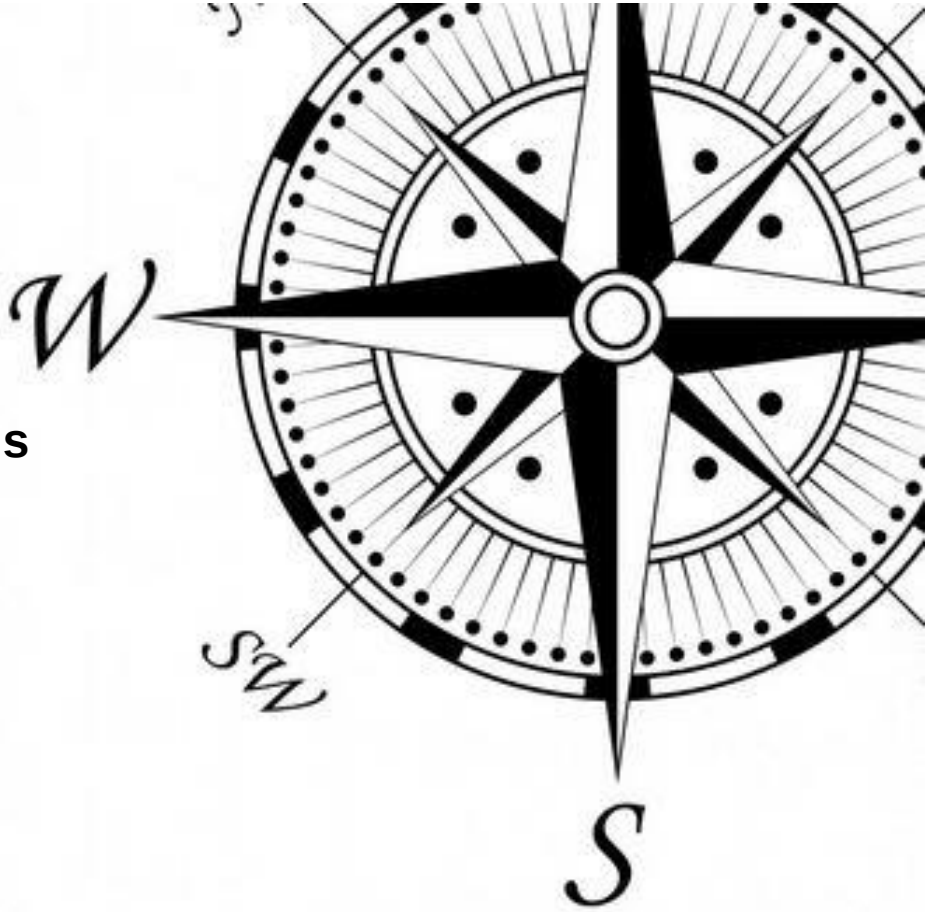
- LMs are pretrained over large amounts **of unstructured and unlabeled text**
  - KBs require manual annotation or complex NLP pipelines to populate

- 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
  - Schema vocabulary mismatch not a big problem

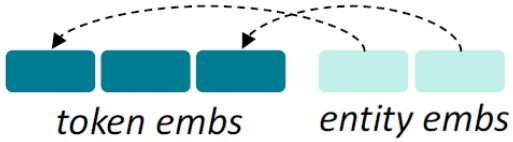
- However, there are also many open challenges to using LMs as KBs:
  - **Hard to interpret** (i.e., why does the LM produce an answer)
  - **Hard to trust** (i.e., the LM may produce a realistic, incorrect answer)
  - **Hard to modify** (i.e., not easy to remove or update knowledge in the LM)

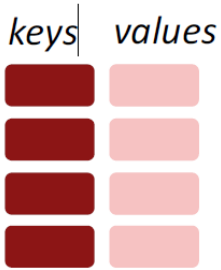
Topics Today

1. What Does a Language Model Know?
2. **Techniques to Add Knowledge to LMs**
3. Evaluating Knowledge in LMs



Techniques to Add Knowledge to LMs





- **Add pretrained entity embeddings**
  - ERNIE
  - KnowBERT
  - **…**
- **Use an external memory**
  - KGLM
  - kNN-LM
  - **…**
- **Modify the training data**
  - WKLM
  - ERNIE (another!)
  - salient span masking
  - …

Method 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.

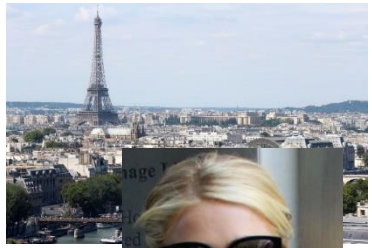
- Pretrained word embeddings do **not** have a notion of entities
  - **Different word embeddings** for “U.S.A.”, “United States of America” and “America” even though these refer to the same entity

- 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!

Reminder: Word Classification

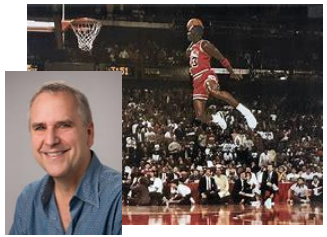
- Meaning of words only clear in context
- Ambigous named entities
  - Paris, Michael Jordan, Orange
  - „Mr. Jordan visited Paris last week“



Mentions

Mentions

Candidates



Candidates

- Entity linking: connect **mention** to **ID**, e.g., from Wikidata
  - -> Chose correct entity embedding for given mention

Method 1: Add Pretrained (Entity) Embeddings

- Entity embeddings are like word embeddings, but for entities in a KB!

- *George Washington* = $\begin{bmatrix} 0.894 \\ 0.693 \\ 0.158 \\ 0.283 \\ 0.143 \\ 0.130 \end{bmatrix}$

- 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)

Method 1: Add Pretrained Entity Embeddings

- Question: How do we incorporate pretrained entity embeddings from a *different embedding space*?

- Answer: Learn a fusion layer to combine context and entity information.

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

- We assume there's a known alignment between entities and words in the sentence such that $e_k = f(w_j)$
  - $\boldsymbol{w}_j$ is the embedding of word j in a sequence of words
  - $\boldsymbol{e}_k$ is the corresponding entity embedding

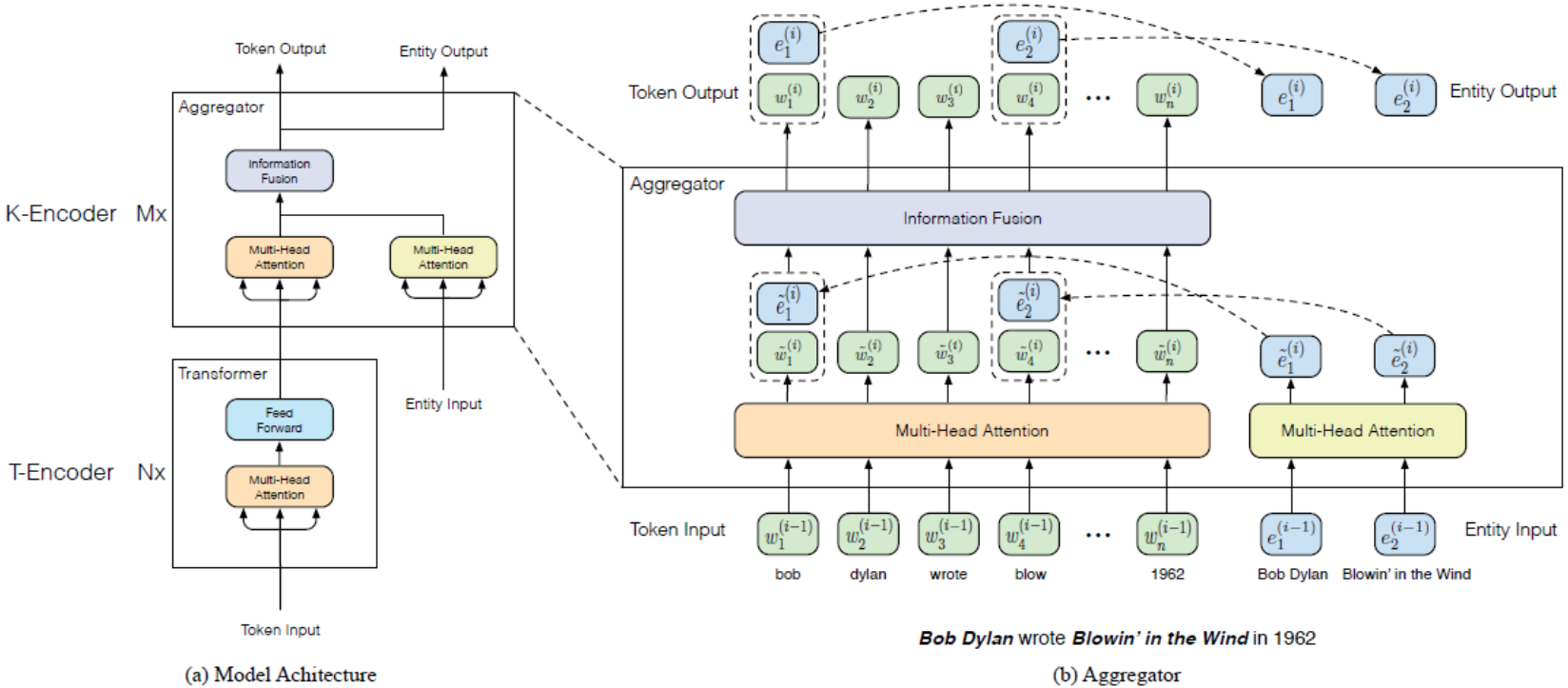
ERNIE

- Enhanced Language Representation with Informative Entities
  - **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
- Pretrain with three tasks:
  - **Masked language model** and **next sentence prediction** (i.e., BERT tasks)
  - Knowledge pretraining task (dEA): randomly mask token-entity alignments and predict corresponding entity for a token from the entities in the sequence
  - $L_{ERNIE} = L_{MLM} + L_{NSP} + L_{dEA}$

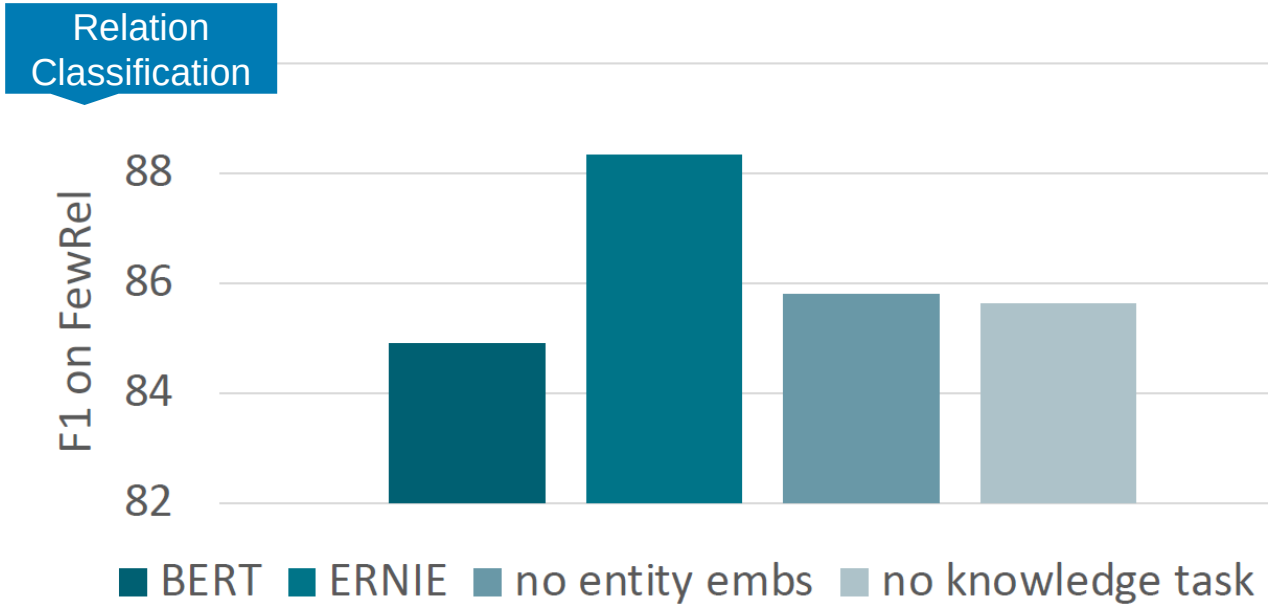
Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., & Liu, Q. (2019). ERNIE: Enhanced Language Representation with Informative Entities. In *ACL* (pp. 1441-1451).

https://i.scdn.co/image/ab6761610000e5ebf4418db5ac929b0de859bfdc

ERNIE


Token Output
Entity Output
Aggregator
Information Fusion
K-Encoder Mx
Multi-Head Attention
Multi-Head Attention
Transformer
Feed Forward
T-Encoder Nx
Multi-Head Attention
Entity Input
Token Input
(a) Model Achitecture
Token Output
Entity Output
Aggregator
Information Fusion
Multi-Head Attention
Multi-Head Attention
Token Input
Entity Input
bob
dylan
wrote
blow
1962
Bob Dylan
Blowin' in the Wind
Bob Dylan wrote Blowin' in the Wind in 1962
(b) Aggregator

ERNIE: Ablation Study



ERNIE: Conclusions

- 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 LM

Method 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

KGLM

- “Barack's Wife Hillary: Using KGs for Fact-Aware Language Modeling”
- Key idea: **condition** the language model on a knowledge graph (KG)
- Recall that language models predict the next word by computing

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

where $x^{(1)}, \dots, x^{(t)}$ is a sequence of words

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

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

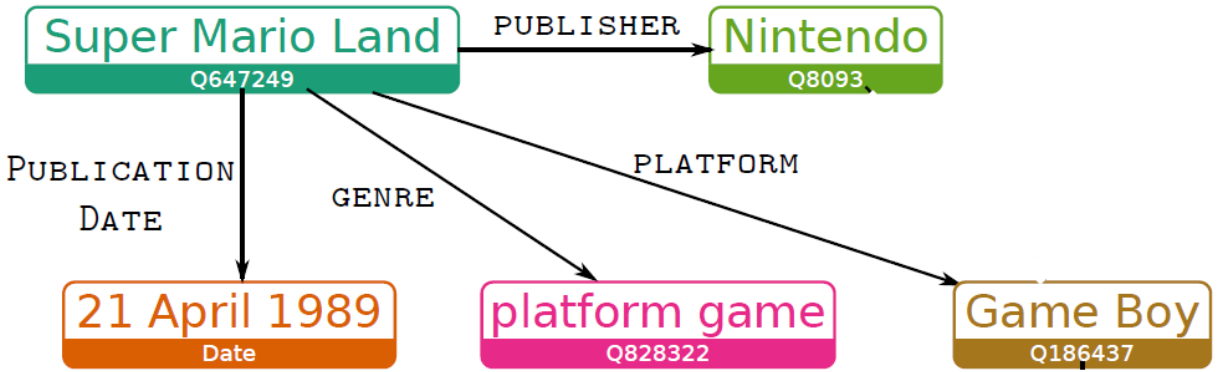
where $\mathcal{E}^{(t)}$ is the set of KG entities mentioned at timestep t

Logan, R., Liu, N. F., Peters, M. E., Gardner, M., & Singh, S. (2019). Barack’s Wife Hillary: Using Knowledge Graphs for Fact-Aware Language Modeling. In *ACL* (pp. 5962-5971).

- Build a **“local” knowledge graph** as you iterate over the sequence
  - Local KG: subset of the full KG with only entities relevant to the sequence

Super Mario Land is a game developed by **Nintendo**.

Assumes entities are known during training!

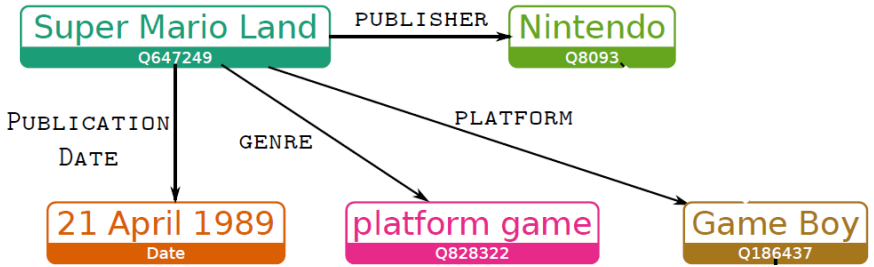


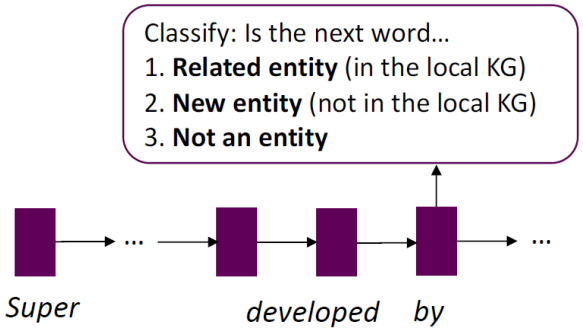
- **When** should the LM **use the local KG** to **predict the next word**?

KGLM

Super Mario Land is a game developed by **Nintendo**.

New entity | Not an entity | Related entity



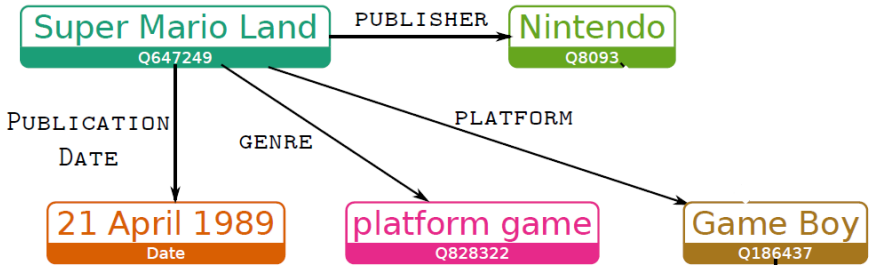


- Use the LSTM hidden state to predict the type of the next word (3 classes)
- **How** does the LM predict the next entity and word in each case?

KGLM

Super Mario Land is a game developed by **Nintendo**.

New entity | Not an entity | Related entity



- Related entity (in the local KG)
- 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

Super Mario Land is a game developed by **Nintendo**.

New entity | Not an entity | Related entity

- Related entity (in the local KG)
  - Find the top-scoring parent and relation in the local KG using the LSTM hidden state and pretrained entity and relation embeddings
    - $P(p_t) = \text{softmax}(\boldsymbol{v_p} \cdot \boldsymbol{h_t})$, where p_t is the "parent" entity, $\boldsymbol{v_p}$ is the corresponding entity embedding, and $\boldsymbol{h_t}$ is from the LSTM hidden state
  - **Next entity**: tail entity from KG triple of (top parent entity, top relation, tail entity)
  - **Next word**: most likely next token over vocabulary expanded to include entity aliases
    - Phrases that could refer to Nintendo (e.g. Nintendo, Nintendo Co., Koppai)

KGLM

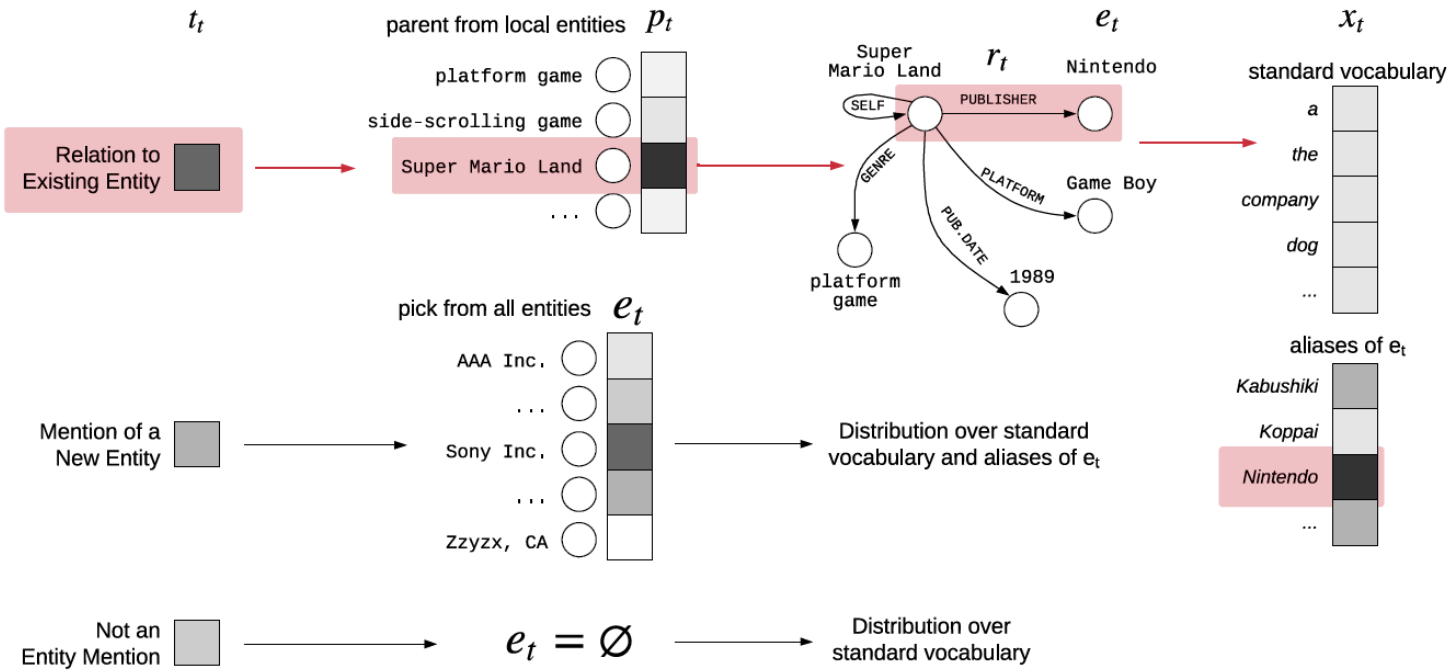
Super Mario Land is a game developed by **Nintendo**.

New entity | Not an entity | Related entity

- New entity (not in the local KG)
  - Find the top-scoring entity in the full KG using the LSTM hidden state and pretrained entity embeddings
  - **Next entity**: directly predict top-scoring entity
  - **Next word**: most likely next token over vocabulary expanded to include entity aliases
- Not an entity
  - **Next entity**: None
  - **Next word**: most likely next token over standard vocabulary

KGLM

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



KGLM

- Outperforms GPT-2 and AWD-LSTM1 on a fact completion task
- Qualitatively, compared to GPT-2, KGLM tends to **predict more specific tokens** (GPT-2 predicts more popular, generic tokens)
- **Supports modifying/updating facts!**
  - Modifying the KG has a direct change in the predictions

Barack Obama was born on _____.

KG triples:	Most likely next word:
(Barack Obama, *birthDate*, 1961-08-04)	“August”, “4”, “1961”
(Barack Obama, *birthDate*, 2013-03-21)	“March”, “21”, “2013”

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 additional memory/computation requirements
  - No modification of the architecture required

WKLM

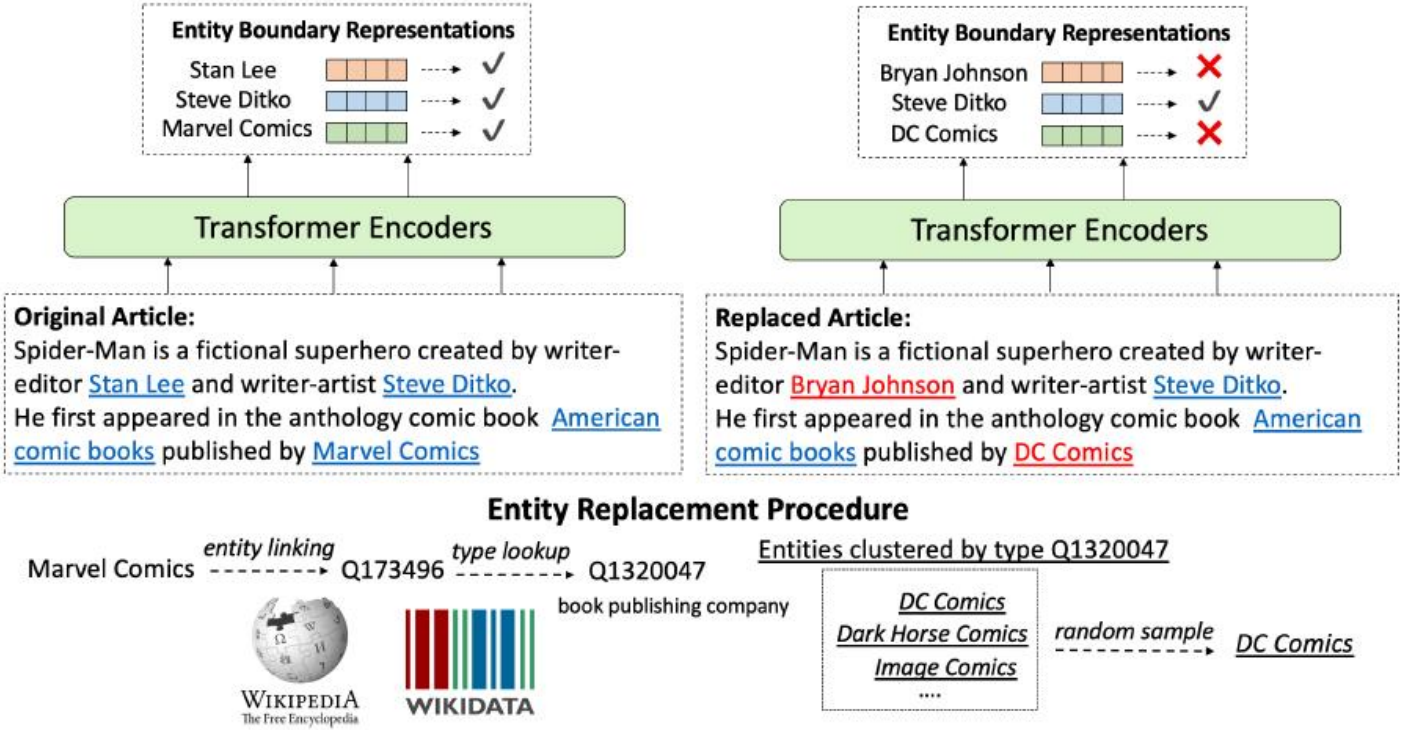
- „Pretrained Encyclopedia: Weakly Supervised Knowledge-Pretrained Language Model“
  - Key idea: train the model to distinguish between true and false knowledge
  - **Replace mentions** in the text with mentions that refer to different entities of the **same type** to create **negative knowledge statements**
    - Model predicts if entity as been replaced or not
    - Type-constraint is intended to enforce linguistically correct sentences

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.

Xiong, W., Du, J., Wang, W. Y., & Stoyanov, V. (2019). Pretrained Encyclopedia: Weakly Supervised Knowledge-Pretrained Language Model. In *ICLR*.

WKLM


Entity Boundary Representations
Stan Lee
Steve Ditko
Marvel Comics
Transformer Encoders
Original Article:
Spider-Man is a fictional superhero created by writer-editor Stan Lee and writer-artist Steve Ditko.
He first appeared in the anthology comic book American comic books published by Marvel Comics
Entity Boundary Representations
Bryan Johnson
Steve Ditko
DC Comics
Transformer Encoders
Replaced Article:
Spider-Man is a fictional superhero created by writer-editor Bryan Johnson and writer-artist Steve Ditko.
He first appeared in the anthology comic book American comic books published by DC Comics
Entity Replacement Procedure
Marvel Comics
entity linking
Q173496
type lookup
Q1320047
book publishing company
WIKIPEDIA
The Free Encyclopedia
WIKIDATA
Entities clustered by type Q1320047
DC Comics
Dark Horse Comics
Image Comics
....
random sample
DC Comics

WKLM

- Uses an entity replacement loss to train the model to distinguish between true and false mentions

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

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**

WKLM

- Improves over BERT and GPT-2 in fact completion tasks
- Improves over ERNIE on a downstream task (entity typing)
- Ablation experiments (QA datasets and one fine-grained NER dataset)
  - MLM loss is essential for downstream task performance
  - WKLM outperforms training longer with just MLM loss

Model	SQuAD		TriviaQA		Quasar-T		FIGER
	EM	F1	EM	F1	EM	F1	Acc
Our BERT	83.4	90.5	48.7	53.2	40.4	46.1	54.53
WKLM	84.3	**91.3**	**52.2**	**56.7**	**43.7**	**49.9**	**60.21**
WKLM without MLM	80.5	87.6	48.2	52.5	42.2	48.1	58.44
WKLM with 15% masking	84.1	91.0	51.0	55.3	42.9	49.0	59.68
Our BERT + 1M MLM updates	**84.4**	91.1	52.0	56.3	42.3	48.2	54.17

Summary: Techniques to Add Knowledge to LMs

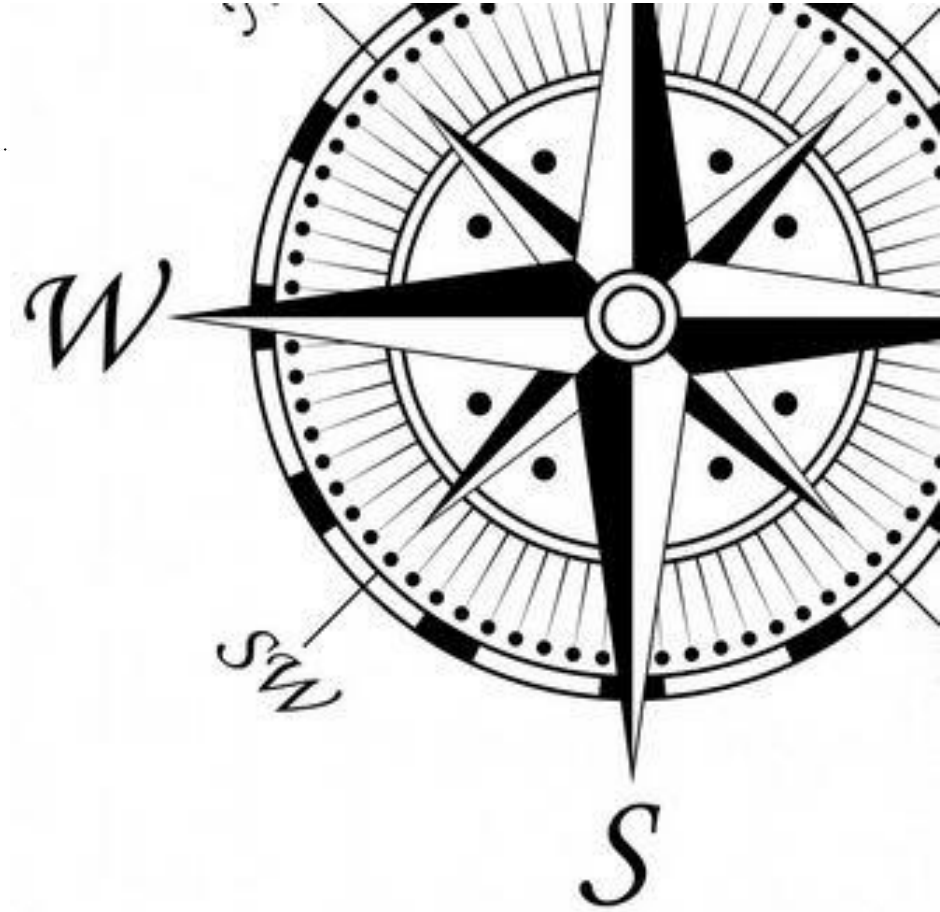
1. Use pretrained entity embeddings
   - Often easy to apply to existing architectures to **leverage KG pretraining**
   - **Indirect way** of incorporating knowledge and can be **hard to interpret**

2. Add an external memory
   - Can support some **updating of factual knowledge** and **easier to interpret**
   - Tend to be more **complex in implementation** and **require more memory**

3. Modify the training data
   - Requires no model changes or additional computation. May also be **easiest to theoretically analyze**! Active area of research
   - Still **open question** if this is always as effective as model changes

Topics Today

1. What Does a Language Model Know?
2. Techniques to Add Knowledge to LMs
3. **Evaluating Knowledge in LMs**



LAnguage Model Analysis (LAMA) Probe

- 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** 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].

Petroni, F., Rocktäschel, T., Riedel, S., Lewis, P., Bakhtin, A., Wu, Y., & Miller, A. (2019). Language Models as Knowledge Bases? In *EMNLP-IJCNLP* (pp. 2463-2473).

LAnguage Model Analysis (LAMA) Probe

- Generate cloze statements from KG triples and question-answer pairs
- Compare LMs to supervised relation extraction (RE) and question answering systems
- **Goal:** evaluate knowledge present in existing pretrained LMs (this means they may have different pretraining corpora!)
- Mean precision at one (P@1)

Corpus	DrQA	RE baseline	fairseq-fconv	Transformer-XL	ELMo	ELMo (5.5B)	BERT-base	BERT-large
Google-RE	-	7.6	2.6	1.6	2.0	3.0	9.8	**10.5**
T-REx	-	**33.8**	8.9	18.3	4.7	7.1	31.1	32.2
ConceptNet	-	-	3.6	5.7	6.1	6.2	15.6	**19.2**
SQuAD	**37.5**	-	3.6	3.9	1.6	4.3	14.1	17.4

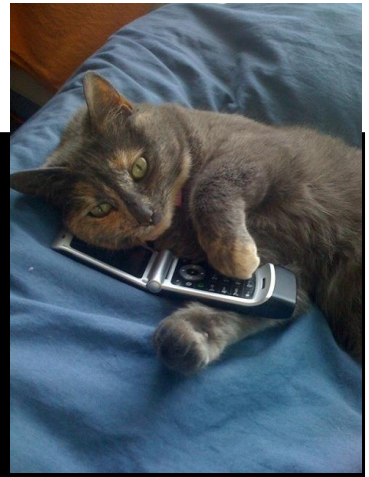
BERT struggles on N-to-M relations

LMs are NOT finetuned!

LAnguage Model Analysis (LAMA) Probe

- You can try out examples to assess knowledge in popular 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

| Perplexity: 2.690

LAnguage Model Analysis (LAMA) Probe

- Limitations of the LAMA probe:
  - Hard to understand **why** models perform well when they do
    - BERT-large may be memorizing co-occurrence patterns rather than “understanding” the cloze statement
    - 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
  - LAMA has only one manually defined template for each relation
  - This means probe results are a **lower bound** on knowledge encoded in the LM

A More Challenging Probe: LAMA-UnHelpful Names (LAMA-UHN)

- Key idea: Remove the examples from LAMA that can be answered without relational knowledge

- Observation: 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, place of birth, nationality, etc.

- BERT's score on LAMA drops ~8% with LAMA-UHN
  - Knowledge-enhanced model E-BERT score 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

Poerner, N., Waltinger, U., & Schütze, H. (2020). E-BERT: Efficient-Yet-Effective Entity Embeddings for BERT. In *EMNLP 2020* (pp. 803-818).

Developing Better Prompts to Query Knowledge in LMs

- LMs may know the fact, but fail on completion tasks due to the query itself
  - Pretraining may be on different contexts and sentence structures than the query

  Example:
  "The birth place of Barack Obama is Honolulu, Hawaii" (pretraining corpus)
  versus
  "Barack Obama was born in _____" (query)

- Generate more LAMA prompts by mining templates from Wikipedia and generating paraphrased prompts by using back-translation

- Ensemble prompts to increase diversity of contexts that fact can be seen in

Jiang, Z., Xu, F. F., Araki, J., & Neubig, G. (2020). How Can We Know What Language Models Know? *Transactions of the Association for Computational Linguistics*, *8*, 423-438.

Developing Better Prompts to Query Knowledge in LMs

- Performance on LAMA for BERT-large increases 7% when using top-performing query for each relation. Ensembling leads to another 4% gain.

- Small changes in the query lead to large gains.
  - LMs are extremely sensitive to the query!

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

Knowledge-Driven Downstream Tasks

- Measures how well the knowledge-enhanced LM transfers its knowledge to **downstream tasks**
- Unlike **probes**, this evaluation usually requires fine-tuning the LM on downstream tasks, such as evaluating BERT on GLUE tasks

- Common tasks:
  - **Relation extraction**
    - Example: *[Bill Gates] was born in [Seattle]; label: city of birth*
  - **Entity typing**
    - Example: *[Alice] robbed the bank; label: criminal*
  - **Question answering**
    - Example: *“What kind of forest is the Amazon?”; label: “moist broadleaf forest”*

Relation extraction performance on TACRED

- Knowledge-enhanced systems (ERNIE, Matching the Blanks, 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
Matching the Blanks (MTB)	BERT-Large	_	_	**71.5**
KnowBert-W+W	BERT-Base	**71.6**	**71.4**	**71.5**

Peters, M. E., Neumann, M., Logan, R., Schwartz, R., Joshi, V., Singh, S., & Smith, N. A. (2019). Knowledge Enhanced Contextual Word Representations. In *EMNLP-IJCNLP* (pp. 43-54).

Entity Typing Performance on Open Entity

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

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

Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., & Liu, Q. (2019). ERNIE: Enhanced Language Representation with Informative Entities. In *ACL* (pp. 1441-1451).
Peters, M. E., Neumann, M., Logan, R., Schwartz, R., Joshi, V., Singh, S., & Smith, N. A. (2019). Knowledge Enhanced Contextual Word Representations. In *EMNLP-IJCNLP* (pp. 43-54).

Recap: Evaluating Knowledge in LMs

- Probes
  - Evaluate the knowledge already present in models without more training
  - Challenging to construct benchmarks that require factual knowledge
  - Challenge to construct the queries used in the probe

- Downstream tasks
  - Evaluate the usefulness of the knowledge-enhanced representation in applications
  - Often requires finetuning the LM further on the downstream task
  - Less direct way to evaluate the knowledge in the LM

Learning Goals for this Chapter

- Understand how and what knowledge can be encoded in language models
- Know techniques to infuse knowledge into language models
- Be able to evaluate LMs for their contained knowledge

- Relevant Chapters:
  - S15 (2021): https://www.youtube.com/watch?v=y68RJVfGoto

Literature

- Petroni, F., Rocktäschel, T., Riedel, S., Lewis, P., Bakhtin, A., Wu, Y., & Miller, A. (2019). Language Models as Knowledge Bases? In *EMNLP-IJCNLP* (pp. 2463-2473).
  - https://arxiv.org/abs/1909.01066
- Roberts, A., Raffel, C., & Shazeer, N. (2020). How Much Knowledge Can You Pack Into the Parameters of a Language Model? In *EMNLP* (pp. 5418-5426).
  - https://arxiv.org/abs/2002.08910
- Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., & Liu, Q. (2019). ERNIE: Enhanced Language Representation with Informative Entities. In *ACL* (pp. 1441-1451).
  - https://arxiv.org/abs/1905.07129
- Peters, M. E., Neumann, M., Logan, R., Schwartz, R., Joshi, V., Singh, S., & Smith, N. A. (2019). Knowledge Enhanced Contextual Word Representations. In *EMNLP-IJCNLP* (pp. 43-54).
  - https://arxiv.org/abs/1909.04164

Literature

- Logan, R., Liu, N. F., Peters, M. E., Gardner, M., & Singh, S. (2019). Barack's Wife Hillary: Using Knowledge Graphs for Fact-Aware Language Modeling. In *ACL* (pp. 5962-5971).
  - https://arxiv.org/abs/1906.07241
- Xiong, W., Du, J., Wang, W. Y., & Stoyanov, V. (2019). Pretrained Encyclopedia: Weakly Supervised Knowledge-Pretrained Language Model. In *ICLR*.
  - https://arxiv.org/abs/1912.09637
- Poerner, N., Waltinger, U., & Schütze, H. (2020). E-BERT: Efficient-Yet-Effective Entity Embeddings for BERT. In *EMNLP 2020* (pp. 803-818).
  - https://arxiv.org/abs/1911.03681
- Jiang, Z., Xu, F. F., Araki, J., & Neubig, G. (2020). How Can We Know What Language Models Know? *Transactions of the Association for Computational Linguistics*, *8*, 423-438.
  - https://aclanthology.org/2020.tacl-1.28/