Knowledge Integration

Natural Language Processing

(based on revision of Megan Leszczynski Lectures)

Announcement

- TA announcements (if any)...

Suggested Readings

1. ERNIE: Enhanced Language Representation with Informative Entities
2. Barack's Wife Hillary: Using Knowledge Graphs for Fact-Aware Language Modeling
3. Pretrained Encyclopedia: Weakly Supervised Knowledge-Pretrained Language Model
4. Language Models as Knowledge Bases?

Motivation

Recap: LMs

- **Standard language models** (LM) 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** (MLM) (e.g., BERT) instead predict a masked token which enables bidirectional context learning

The ***[MASK]*** *opened their* ***[MASK]*****.**

- Both are great for training over large amounts of unlabeled text!

Recap: LMs

- 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**?

What does a LM know?

- 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**.

Examples taken from **Petroni et al., EMNLP 2019** to test BERT-Large

Language Models as Knowledge Bases? Petroni et al., 2019, https://aclanthology.org/D19-1250.pdf

What does a LM know?

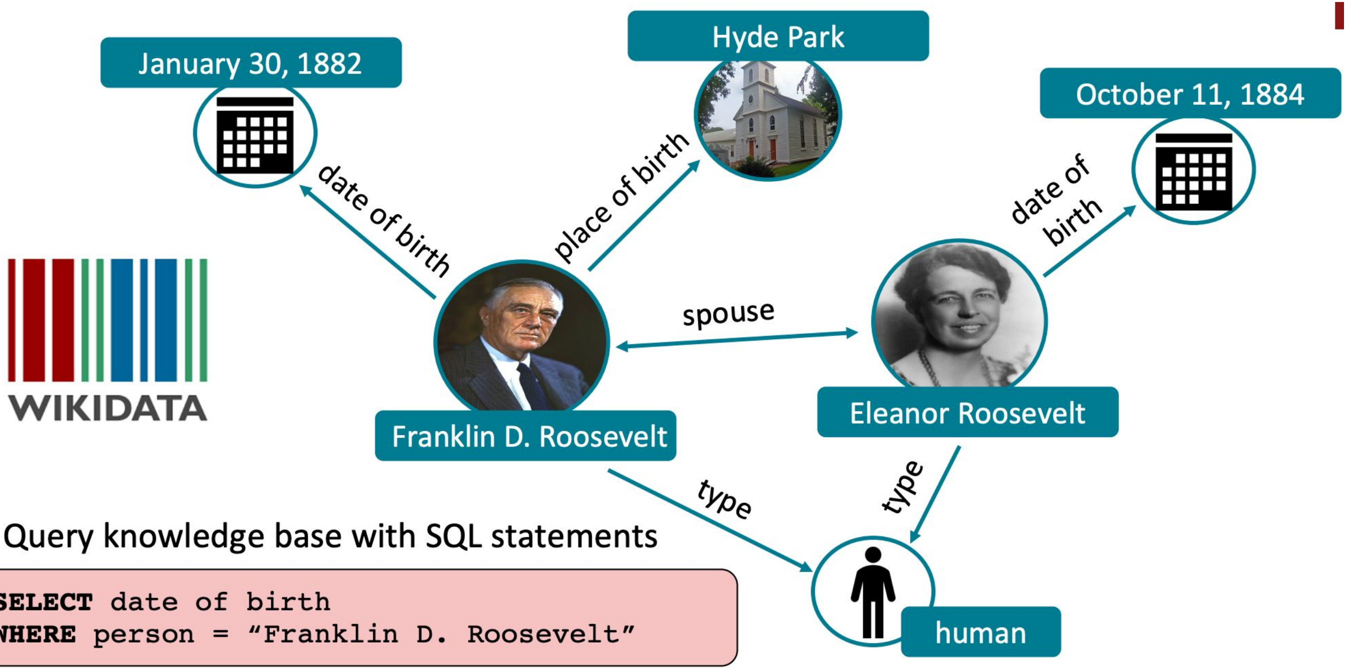
- Predictions generally make sense (e.g. the correct types), but are not all factually correct.
- Why might this happen?
  - **Unseen facts**: some facts may not have occurred in the training corpora at all
  - **Rare facts**: LM hasn't seen enough examples during training to memorize the fact
  - **Model 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 have a way to go.

The importance of knowledge

- 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 (i.e., entity relation tasks)

- **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!

Knowledge graph as knowledge bases

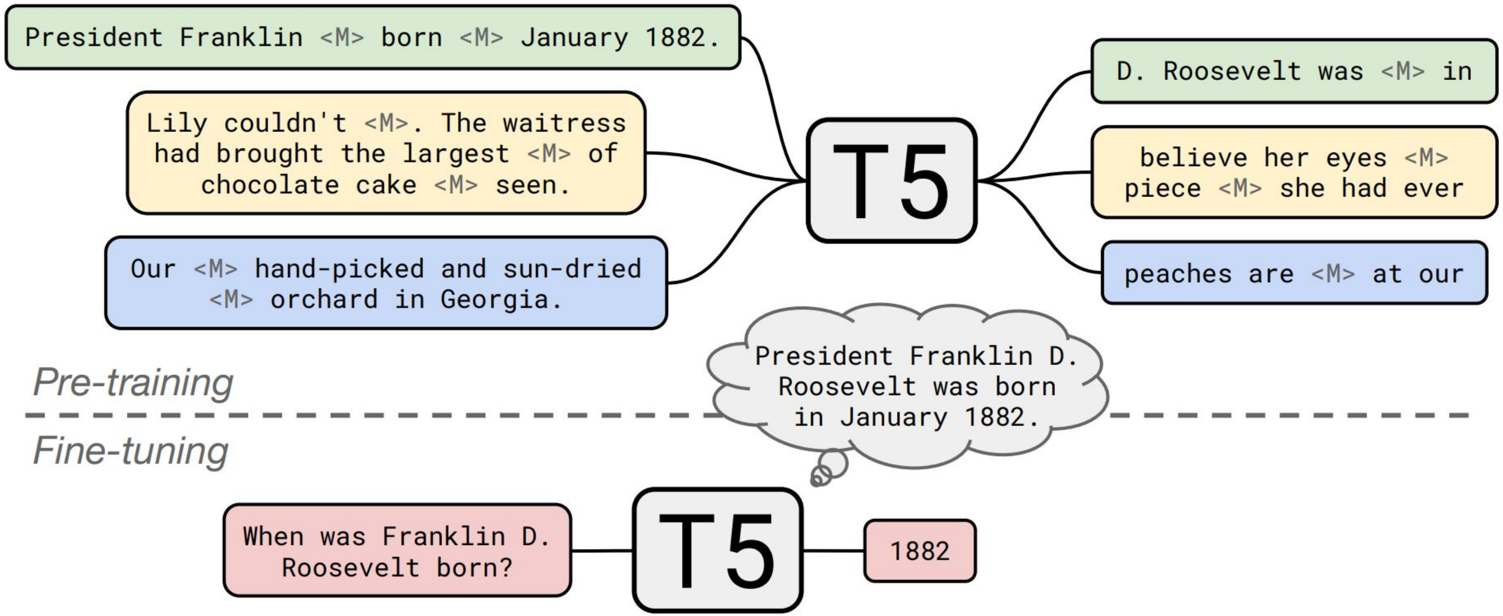


- Query knowledge base with SQL statements

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

A knowledge consists of **head entity**, **tail entity** and the **relations**, creating what we called the **knowledge triples, e.g.,** Franklin (head), January 30,1882 (tail), date of birth (relations)

LM as knowledge bases



LM vs. 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)

Knowledge Integration Techniques

Knowledge Integration Techniques

- **Add pretrained entity embeddings**
  - Idea: combine the embeddings of the entity
    - ERNIE
    - KnowBERT
- **Use an external memory**
  - Idea: use external KBs for helping the prediction
    - KGLM
    - kNN-LM
- **Modify the training data**
  - Idea: modify training objective to better learn knowledge
    - WKLM
    - ERNIE (not the same as above), 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
  - For example, 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”

Method 1: Add pretrained entity embeddings

Entity embeddings can be useful to LMs iff you can do **entity linking** well!

mention   *mention*

<u>Washington</u> was the first president of the <u>United States</u>.

candidate

Q23 (Wikidata)

candidate

Q1223 (Wikidata)

candidate

Q30 (Wikidata)

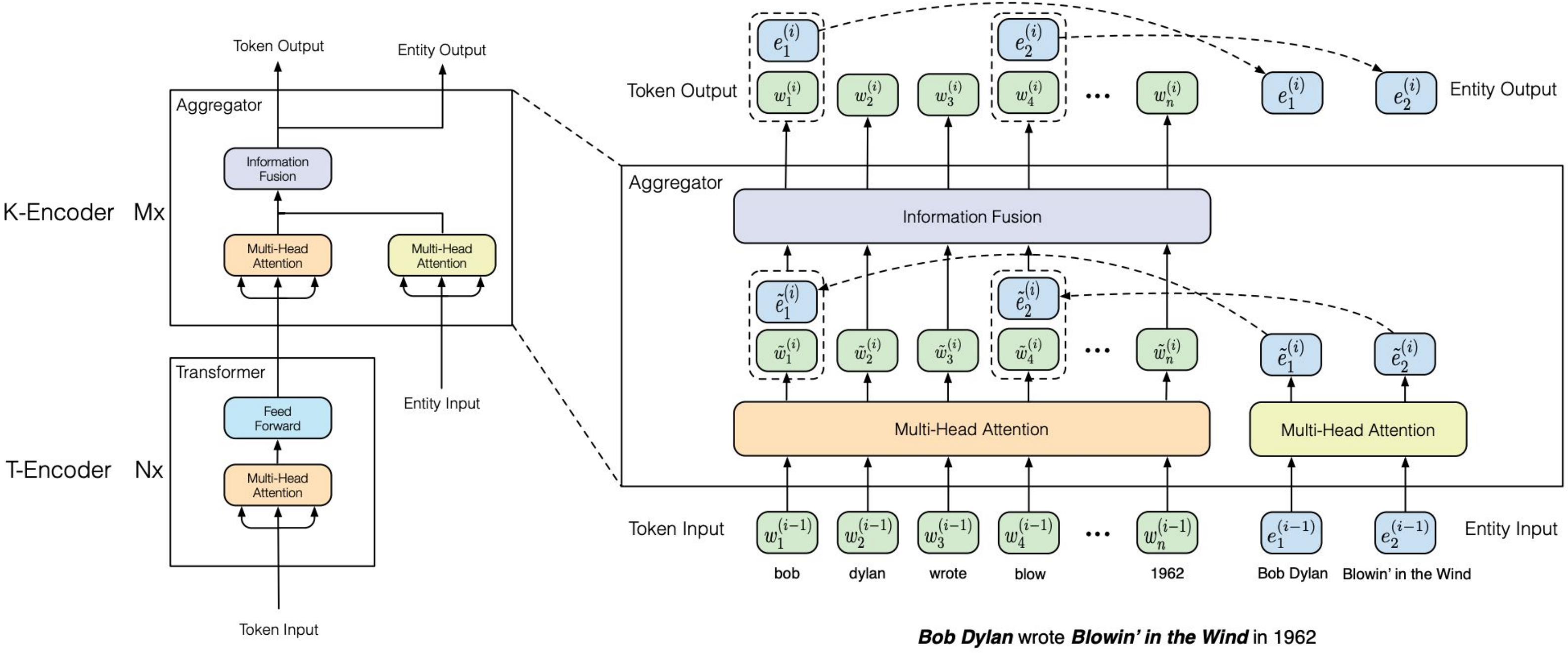
Bootleg: Chasing the Tail with Self-Supervised Named Entity Disambiguation, Orr et al. 2021, http://cidrdb.org/cidr2021/papers/cidr2021_paper13.pdf
Efficient One-Pass End-to-End Entity Linking for Questions, Li et al., 2020, https://arxiv.org/pdf/2010.02413.pdf

Method 1: Add pretrained entity embeddings

- Entity embeddings are just like word embeddings, but for entities in a knowledge base
- Many techniques for training entity embeddings:
  - **TransE**
    - Model is trained so that Subject + Relationships = Object
  - **Wikipedia2Vec**
    - Train just like the skipgram model, where model is trained to predict the connected entities given a entity
  - Transformer encodings of entity descriptions (e.g., **BLINK**)
    - Use two BERT encoders, one for **encoding the independent entity embedding**, and **another encoder for linking the two entities**

Translating Embeddings for Modeling Multi-relational Data, Bordes et al. 2013, https://papers.nips.cc/paper/2013/hash/1cecc7a77928ca8133fa24680a88d2f9-Abstract.html
Wikipedia2Vec: An Efficient Toolkit for Learning and Visualizing the Embeddings of Words and Entities from Wikipedia, Yamada et al. 2020, https://aclanthology.org/2020.emnlp-demos.4.pdf
Scalable Zero-shot Entity Linking with Dense Entity Retrieval, Wu et al., 2020, https://arxiv.org/pdf/1911.03814v3.pdf

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



(a) Model Achitecture

(b) Aggregator

ERNIE: Enhanced Language Representation with Informative Entities, Zhang et al. 2019, https://arxiv.org/pdf/1905.07129.pdf

ERNIE: Enhanced Language Representation with Informative Entities

Performances on entity typing (using FIGER and Open Entity dataset)

Model	Acc.	Macro	Micro
NFGEC (Attentive)	54.53	74.76	71.58
NFGEC (LSTM)	55.60	75.15	71.73
BERT	52.04	75.16	71.63
ERNIE	**57.19**	**76.51**	**73.39**

Table 2: Results of various models on FIGER (%).

Model	P	R	F1
NFGEC (LSTM)	68.80	53.30	60.10
UFET	77.40	60.60	68.00
BERT	76.37	70.96	73.56
ERNIE	**78.42**	**72.90**	**75.56**

Table 3: Results of various models on Open Entity (%).

ERNIE: Enhanced Language Representation with Informative Entities

Performances on entity relations classification (using FewRel and TACRED dataset)

Model	FewRel			TACRED		
	P	R	F1	P	R	F1
CNN	69.51	69.64	69.35	70.30	54.20	61.20
PA-LSTM	-	-	-	65.70	64.50	65.10
C-GCN	-	-	-	69.90	63.30	66.40
BERT	85.05	85.11	84.89	67.23	64.81	66.00
ERNIE	88.49	88.44	**88.32**	69.97	66.08	**67.97**

Table 5: Results of various models on FewRel and TACRED (%).

ERNIE: Enhanced Language Representation with Informative Entities

Strengths:

- Combines entity + context info through fusion layers
  - Why not just concatenate?
    - Authors didn't mention clearly why, but it could be the magnitude of the embeddings values interfering with...
  - Improves **downstream knowledge-driven** tasks

Limitations

- Require text data with **entities annotated as input**, i.e., not really a end-to-end architecture

KnowBERT [Peters et al. 2019]

- Key idea: pretrain an integrated entity linker (EL) as an extension to BERT

$$L_{knowBERT} = L_{NSP} + L_{MLM} + \mathbf{L_{EL}}$$

- On downstream tasks, **EL predicts entities so entity annotations aren't required**
  - Hopefully, learning EL can better encode knowledge - shows performance gains over ERNIE on downstream tasks

- Like ERNIE, knowBERT uses a fusion layer to combine entity and context information and adds a knowledge pretraining task

Knowledge Enhanced Contextual Word Representations, Peters et al. 2019, https://aclanthology.org/D19-1005.pdf

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) and **simply copy them when they are applicable**

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

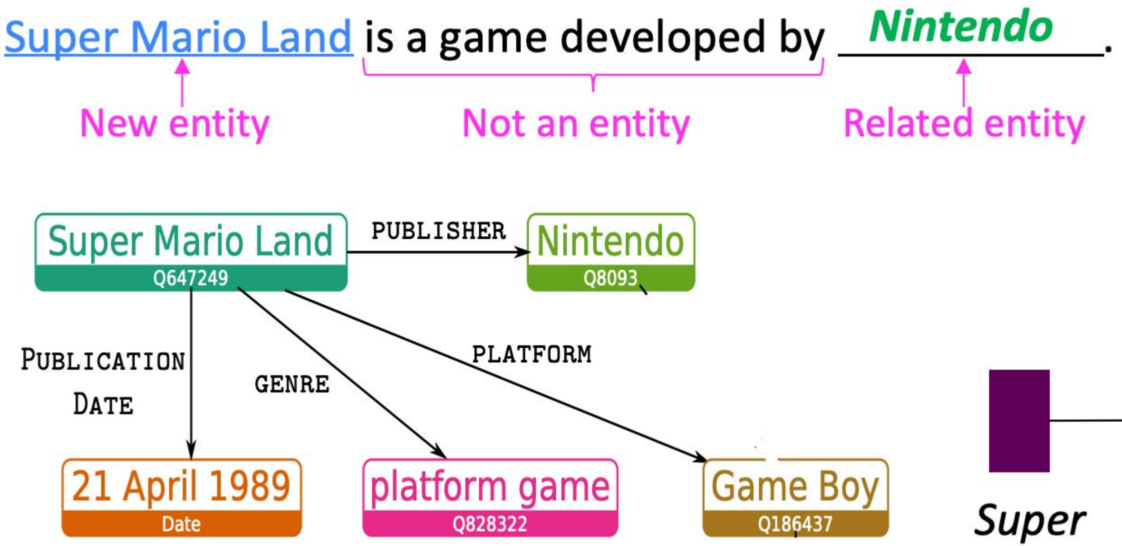
KGLM [Logan et al., ACL 2019]

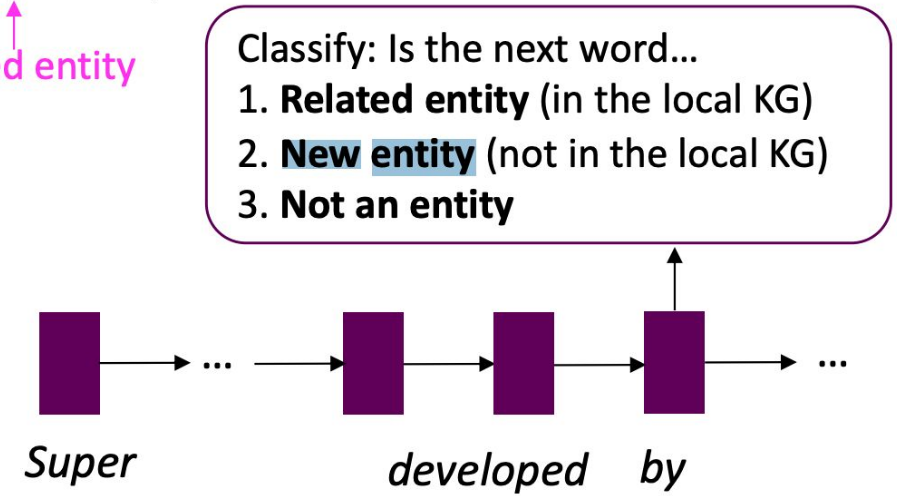
- Idea: Run over the sequence. Train the model so that it is able to know whether is a (1) new entity, (2) existing entity, and (3) not an entity

- If it is a **new entity**, look through the full knowledge graph and then find all the possible aliases and then determine the best answer
  - Add this information to a dynamically growing local graph
- If it is an **existing entity**, look up the local knowledge graph to save computation, and see which relation makes most sense
- If it is **not an entity**, simply do the usual distribution over all vocabularies

Barack's Wife Hillary: Using Knowledge Graphs for Fact-Aware Language Modeling, Logan et al., 2019, https://aclanthology.org/P19-1598/

KGLM


Super Mario Land is a game developed by Nintendo.
New entity
Not an entity
Related entity
Super Mario Land
Q647249
PUBLISHER
Nintendo
Q8093
PUBLICATION DATE
GENRE
PLATFORM
21 April 1989
Date
platform game
Q828322
Game Boy
Q186437


Classify: Is the next word...
1. Related entity (in the local KG)
2. New entity (not in the local KG)
3. Not an entity
...
...
Super
developed
by

KGLM

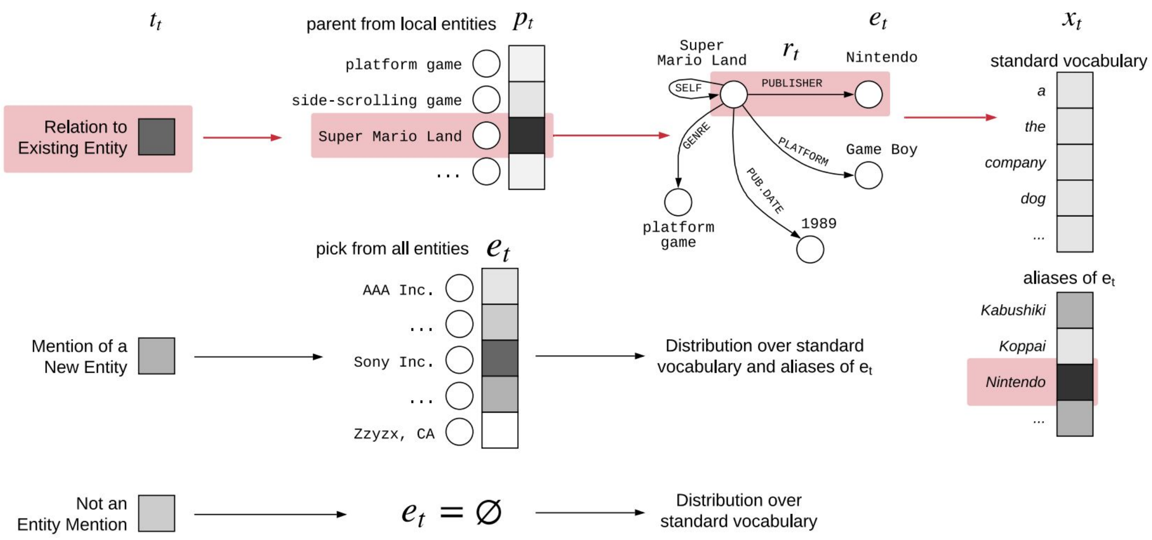


Figure 2: **KGLM Illustration.** When trying to generate the token following "*published by*", the model first decides the type of the mention (t_t) to be a related entity (darker indicates higher probability), followed by identifying the parent (p_t), relation (r_t), and entity to render (e_t) from the local knowledge graph as (Super Mario Land, *Publisher*, Nintendo). The final distribution over the words includes the standard vocabulary along with aliases of Nintendo, and the model selects "*Nintendo*" as the token x_t. Facts related to Nintendo will be added to the local graph.

KGLM

- Outperforms GPT-2 and AWD-LSTM on a fact completion task
- Qualitatively, compare 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

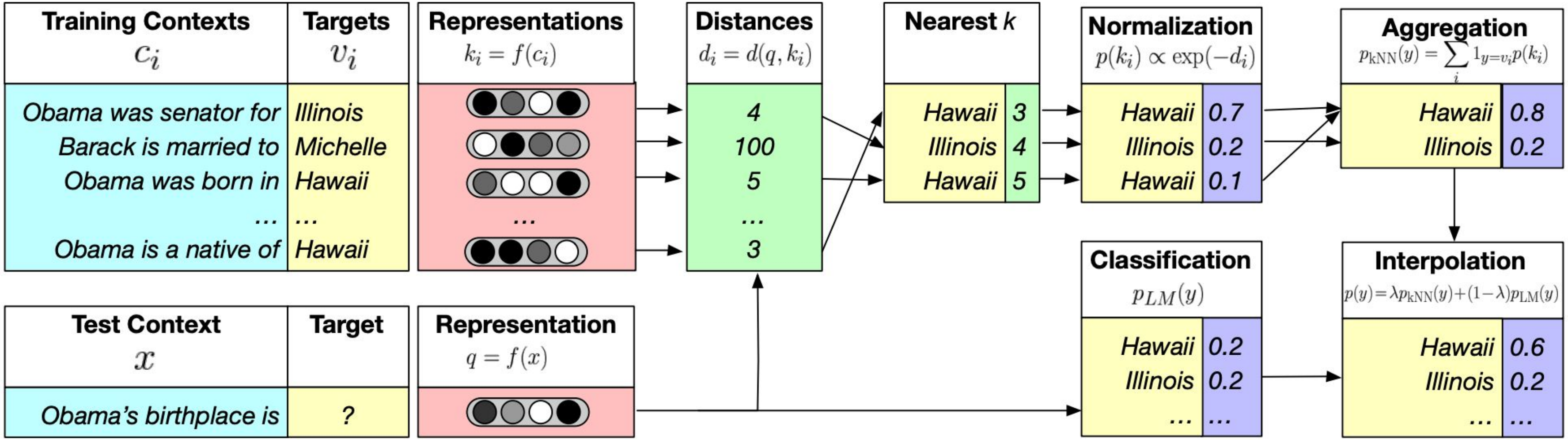
Nearest Neighbor LM [Khandelwal et al., ICLR 2020]

- Idea: **learning the similarities between text sequences** is easier than predicting the next word
  - Example: “Dickens is the author of _____” = “Dickens wrote ______”
- So, store all representations of text sequences in a nearest neighbor datastore!
- At inference:
  - Find the k most similar sequences of text in the datastore
  - Retrieve the corresponding values (i.e., the next word) for the k sequences
  - Combine the **kNN probabilities and LM probabilities** for the final prediction

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

Generalization through Memorization: Nearest Neighbor Language Models, Khandelwal et al., 2020, https://arxiv.org/pdf/1911.00172.pdf

Nearest Neighbor LM



Note that *f* here is a transformer-based model, *d* here is simple L^2 distance

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

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

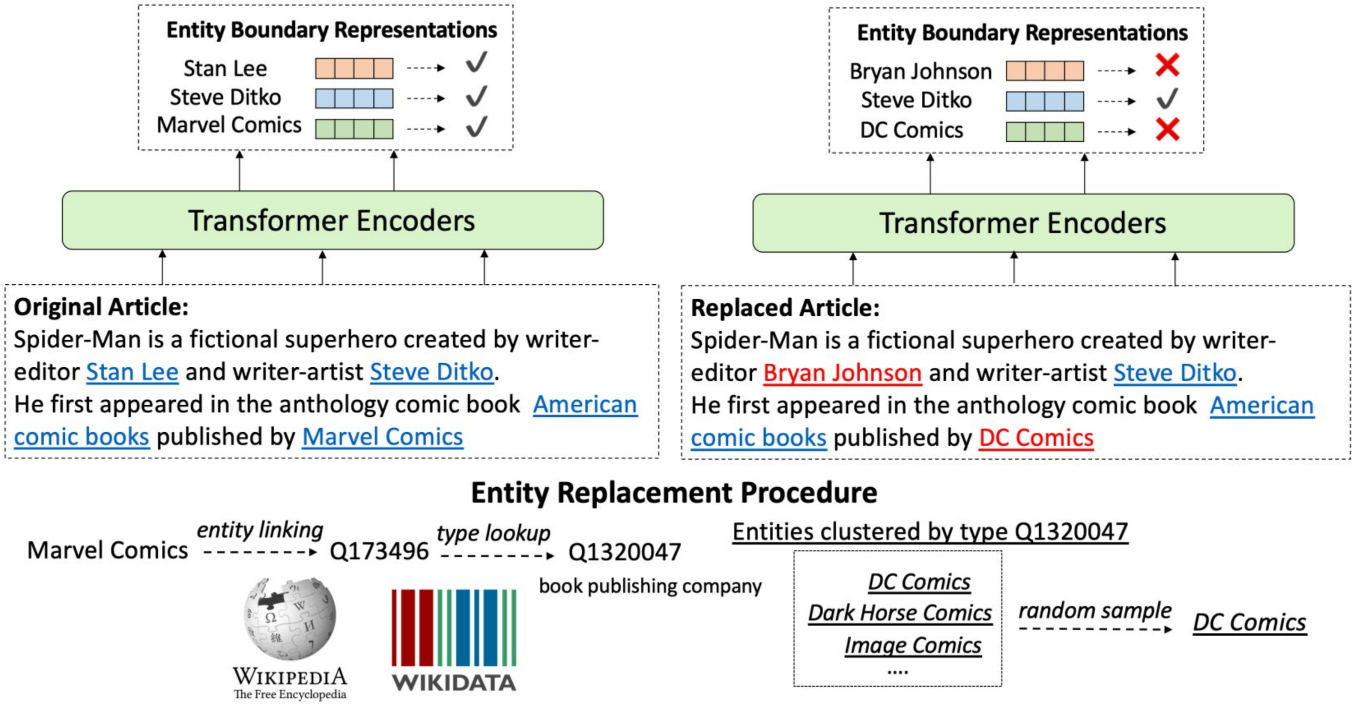
- **Replace mentions** in the text with mentions that refer to different entities of the **same type** to create **negative knowledge statements**
  - Model predicts if entity has been replaced
  - Type-constraint is intended to enforce linguistically correct sentences

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

Pretrained Encyclopedia: Weakly Supervised Knowledge-Pretrained Language Model, Xiong et al., https://arxiv.org/pdf/1912.09637.pdf

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



Pretrained Encyclopedia: Weakly Supervised Knowledge-Pretrained Language Model, Xiong et al., https://arxiv.org/pdf/1912.09637.pdf

Pretrained Encyclopedia: Weakly Supervised Knowledge-Pretrained Language Model (WKLM)

- Improved over BERT and GPT-2 in fact completion tasks
- Improves over ERNIE on a downstream task (entity typing)
- Ablation experiments (i.e., removing some components experiment)
    - MLM loss is essential for downstream task performance
    - WKLM outperforms training 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

ERNIE: Enhanced Representation through Knowledge Integration [Sun et al., arXiv 2019] (another ERNIE paper)

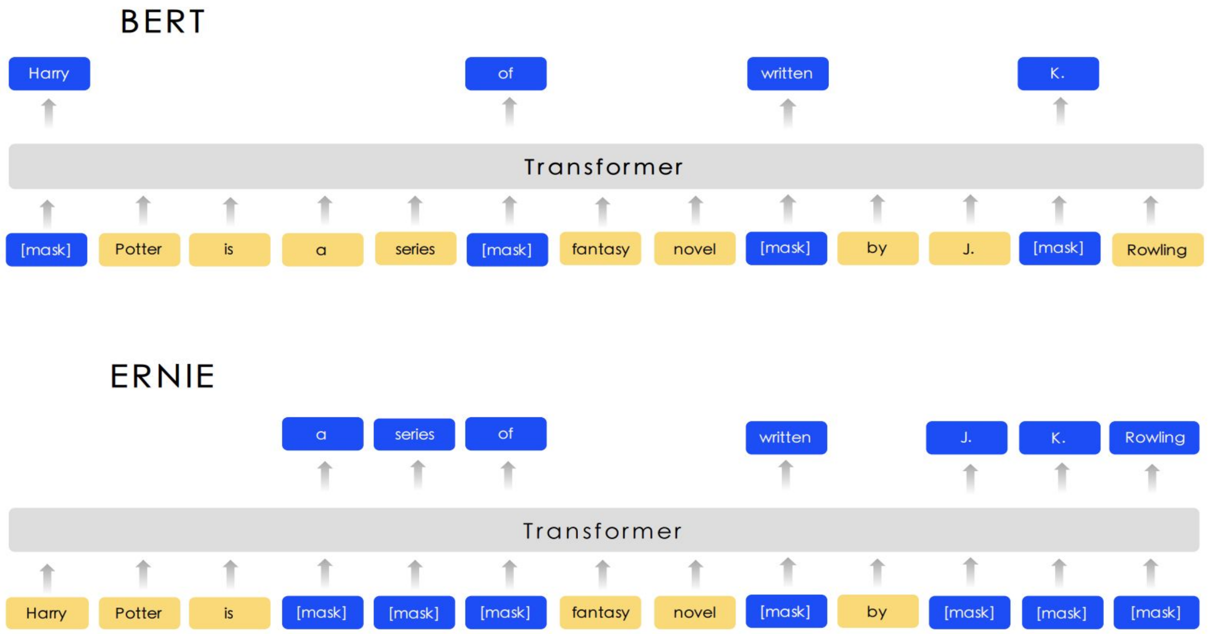


Figure 1: The different masking strategy between BERT and ERNIE

ERNIE: Enhanced Representation through Knowledge Integration, Sun et al., https://arxiv.org/abs/1904.09223

REALM: Retrieval-Augmented Language Model Pre-Training [Guu et al., ICML 2020]

Salient span masking - focus on examples that require world knowledge to predict the masked tokens (e.g., location, date)

Name	Architectures	Pre-training	NQ (79k/4k)	WQ (3k/2k)	CT (1k /1k)	# params
BERT-Baseline (Lee et al., 2019)	Sparse Retr.+Transformer	BERT	26.5	17.7	21.3	110m
T5 (base) (Roberts et al., 2020)	Transformer Seq2Seq	T5 (Multitask)	27.0	29.1	-	223m
T5 (large) (Roberts et al., 2020)	Transformer Seq2Seq	T5 (Multitask)	29.8	32.2	-	738m
T5 (11b) (Roberts et al., 2020)	Transformer Seq2Seq	T5 (Multitask)	34.5	37.4	-	11318m
DrQA (Chen et al., 2017)	Sparse Retr.+DocReader	N/A	-	20.7	25.7	34m
HardEM (Min et al., 2019a)	Sparse Retr.+Transformer	BERT	28.1	-	-	110m
GraphRetriever (Min et al., 2019b)	GraphRetriever+Transformer	BERT	31.8	31.6	-	110m
PathRetriever (Asai et al., 2019)	PathRetriever+Transformer	MLM	32.6	-	-	110m
ORQA (Lee et al., 2019)	Dense Retr.+Transformer	ICT+BERT	33.3	36.4	30.1	330m
Ours ($\mathcal{X}$ = Wikipedia, $\mathcal{Z}$ = Wikipedia)	Dense Retr.+Transformer	REALM	39.2	40.2	**46.8**	330m
Ours ($\mathcal{X}$ = CC-News, $\mathcal{Z}$ = Wikipedia)	Dense Retr.+Transformer	REALM	**40.4**	**40.7**	42.9	330m

REALM: Retrieval-Augmented Language Model Pre-Training, Guu et al., https://arxiv.org/pdf/2002.08909.pdf

REALM: Retrieval-Augmented Language Model Pre-Training

Through the **ablation experiments**, it is clear that span salient masking creates a huge difference (REALM vs. REALM with random uniform masks vs. random span masks)

Table 2. Ablation experiments on NQ's development set.

Ablation	**Exact Match**	**Zero-shot Retrieval Recall@5**
REALM	38.2	38.5
REALM retriever+Baseline encoder	37.4	38.5
Baseline retriever+REALM encoder	35.3	13.9
Baseline (ORQA)	31.3	13.9
REALM with random uniform masks	32.3	24.2
REALM with random span masks	35.3	26.1
30× stale MIPS	28.7	15.1

Evaluating knowledge in LMs

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

- How much relational (commonsense and factual) knowledge is already in off-the-shelf language models **without further fine-tuning.**
- Manually constructed a set of what the authors called **cloze statements**
  - basically the same as salient span masking
- Authors tested on using several datasets - **Google-RE** (60K facts from Wikipedia), **T-REx** (subset of Wikidata triples), **ConceptNet** (multilingual base), **SQuAD** (popular question answering dataset)

Corpus	Relation	Statistics #Facts	Statistics #Rel	Baselines Freq	Baselines DrQA	KB RE_n	KB RE_o	LM Fs	LM Txl	LM Eb	LM E5B	LM Bb	LM Bl
Google-RE	birth-place	2937	1	4.6	-	3.5	13.8	4.4	2.7	5.5	7.5	14.9	**16.1**
	birth-date	1825	1	1.9	-	0.0	**1.9**	0.3	1.1	0.1	0.1	1.5	1.4
	death-place	765	1	6.8	-	0.1	7.2	3.0	0.9	0.3	1.3	13.1	**14.0**
	Total	5527	3	4.4	-	1.2	7.6	2.6	1.6	2.0	3.0	9.8	**10.5**
T-REx	1-1	937	2	1.78	-	0.6	10.0	17.0	36.5	10.1	13.1	68.0	**74.5**
	N-1	20006	23	23.85	-	5.4	**33.8**	6.1	18.0	3.6	6.5	32.4	34.2
	N-*M*	13096	16	21.95	-	7.7	**36.7**	12.0	16.5	5.7	7.4	24.7	24.3
	Total	34039	41	22.03	-	6.1	**33.8**	8.9	18.3	4.7	7.1	31.1	32.3
ConceptNet	Total	11458	16	4.8	-	-	-	3.6	5.7	6.1	6.2	15.6	**19.2**
SQuAD	Total	305	-	-	**37.5**	-	-	3.6	3.9	1.6	4.3	14.1	17.4

Table 2: Mean precision at one (P@1) for a frequency baseline (Freq), DrQA, a relation extraction with naïve entity linking (RE_n), oracle entity linking (RE_o), fairseq-fconv (Fs), Transformer-XL large (Txl), ELMo original (Eb), ELMo 5.5B (E5B), BERT-base (Bb) and BERT-large (Bl) across the set of evaluation corpora.

Language Models as Knowledge Bases? Petroni et al., 2019, https://aclanthology.org/D19-1250.pdf

LAnguage Model Analysis (LAMA) Probe

- **Limitations**:
  - Hard to understand **why models perform well** when they do
    - BERT-large may be **memorizing**, NOT **understanding / knowing**!
  - LM is very sensitive to the **phrasing** of the statement
    - LAMA has only one manually defined templates for each relation
    - This means probe results are a lower bound on knowledge encoded in the LM
- **Solution**: Remove examples from LAMA that can be answered without relational knowledge (e.g., "Pope Francis is a pope") [Poerner et al., EMNLP 2020) - (a.k.a LAMA-UHN)
  - Knowledge-enhanced model score drops only <1%
  - BERT's score drops even more (8%) with LAMA-UHN

Knowledge-driven downstream tasks [Peters et al., ACL 2019]

- The previous method (e.g., LAMA) is very intrinsic, we can also look at more extrinsic evaluation….
- Measures how well the knowledge-enhanced LM **transfer its knowledge** to downstream tasks
- The **bad news** is that unlike probes, this evaluation usually requires finetuning the LM on downstream tasks.
- Common tasks for assessing knowledge:
    - **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"

Knowledge Enhanced Contextual Word Representations, Peters et al. 2019, https://aclanthology.org/D19-1005.pdf

Knowledge-driven downstream tasks

Results on testing some models on **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**

Knowledge-driven downstream tasks

- Results on testing some models on **entity typing**
- Impressively, NFGEC and UFET were specifically 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**

Summary

- Use pretrained entity embeddings
  - Often not too difficult to apply
  - Often requires extra pretraining...
  - **Indirect** way of incorporating knowledge and can be **hard to interpret**
- Add an external memory
  - Can support some **updating of factual knowledge** and **easier to interpret**
  - Tend to be more **computationally expensive**
- Modify the training data
  - Requires no model changes or additional computation. Easiest to analyze. **Active area of research.**
  - Still questioning whether the model is memorizing vs. knowing....
- Evaluation
  - Still questioning a “good” benchmark for assessing knowledge
- There are many more papers I haven’t covered....so keep an eye out for all these updates!

Still more!

- Yet another entity embedding for knowledge - E-BERT
  - Poerner et al., EMNLP 2020
- Retrieval-augmented language models
  - REALM, Guu et al., ICML 2020
- Modifying knowledge in language models
  - Modifying Memories in Transformer Models, Zhu et al., arXiv 2020
- More multitask pre-training for language models
  - KEPLER, Wang et al., TACL 2020
- More efficient knowledge systems
  - NeurIPS Efficient QA challenge
- Better knowledge benchmarks
  - KILT, Petroni et al., arXiv 2020