

# Incorporating Knowledge Graphs into Large Language Models

25.01.2024

Nicolaus-Herman Schlegel

Supervisor: Ferdinand Schlatt

# Motivation

## Today's Problems:

- ❑ LMs are black boxes, LM embeddings aren't interpretable
- better insights are needed, interpretation for LM embedding would be helpful
- ❑ language input – language output; what if we want to use output for further computations?
- machine-readable output would be useful

### Example:

The capital of France is [MASK].

There is an Eiffel Tower in [MASK], Tennessee.

→ Paris

What if we want to match Paris with an entity (e.g. from Wikidata)?

- ❑ multiple Paris entities → need for context
- would be nice to also get an entity as output

#### Places [\[ edit \]](#)

##### Canada [\[ edit \]](#)

- [Paris, Ontario](#), a community
- [Paris, Yukon](#), a former community

##### Indonesia [\[ edit \]](#)

- [Paris, Gorontalo](#), a village in [Gorontalo Regency](#)
- [Paris, Highland Papua](#), a village in [Highland Papua](#)

##### United States [\[ edit \]](#)

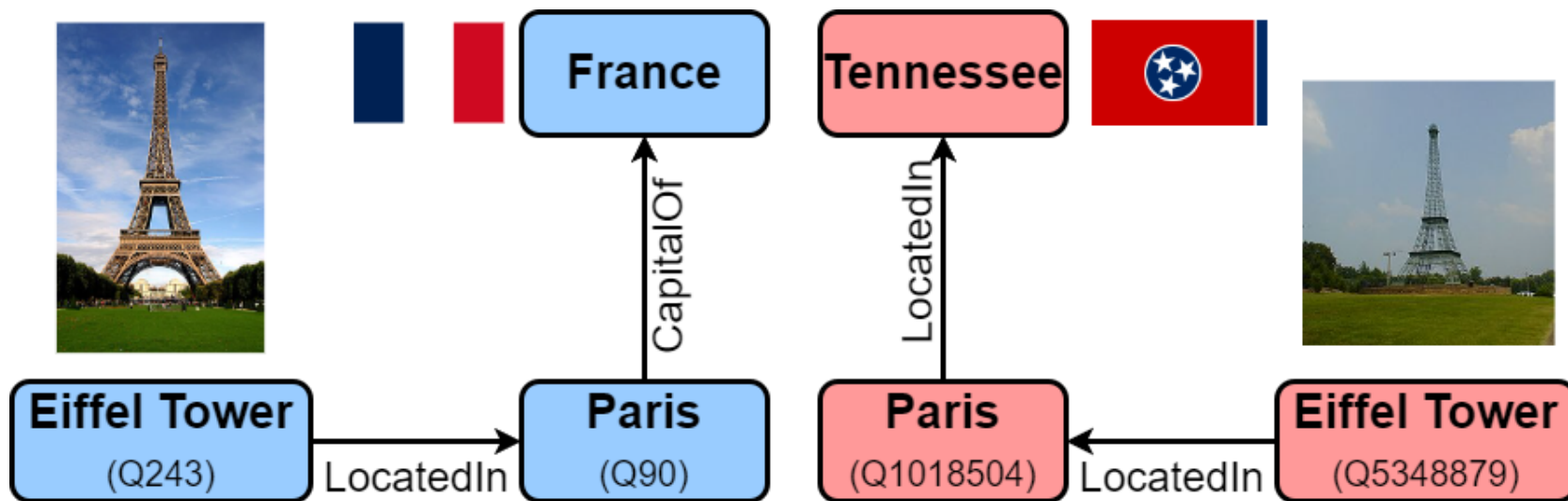
- [Paris, Arkansas](#), a city
- [Paris, Idaho](#), a city
- [Paris, Illinois](#), a city
- [Paris, Indiana](#), an unincorporated community
- [Paris, Iowa](#), an unincorporated community
- [Paris, Kentucky](#), a city
- [Paris, Maine](#), a town

# Idea

## Incorporating Knowledge Graphs

### Knowledge Graphs (KGs)

- represent knowledge (relations between entities)
- in machine-readable format → allows automatic reasoning
- embedding needed for more complex tasks (e.g. link prediction) → can be used in LM



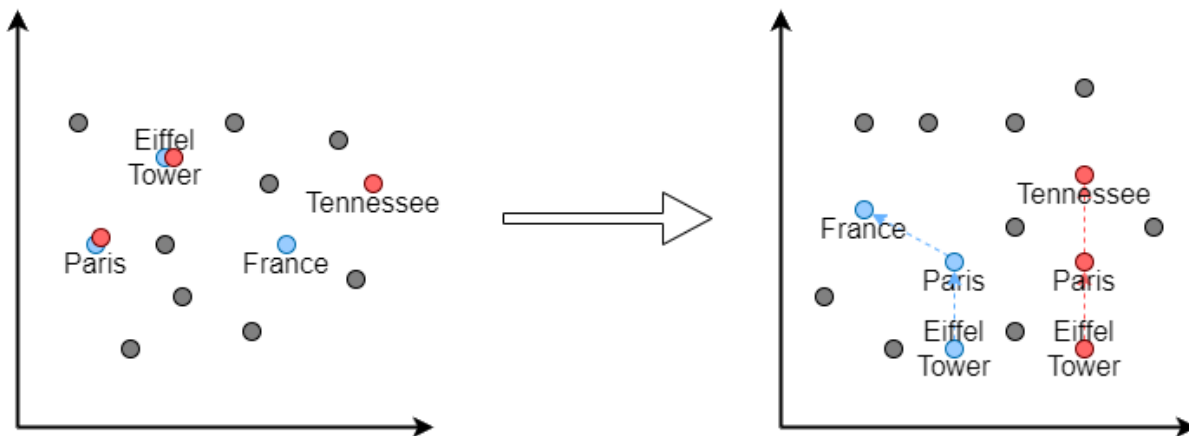
# Idea

## ...into Large Language Models

- ❑ use annotated Wikipedia abstracts (with linked entities + relations) → T-REx Dataset
- ❑ train PTM (BERT) together with KG, use combined KG- and LM-loss
- ❑ fit KG into same vectorspace as LM uses for token embedding

### Goal:

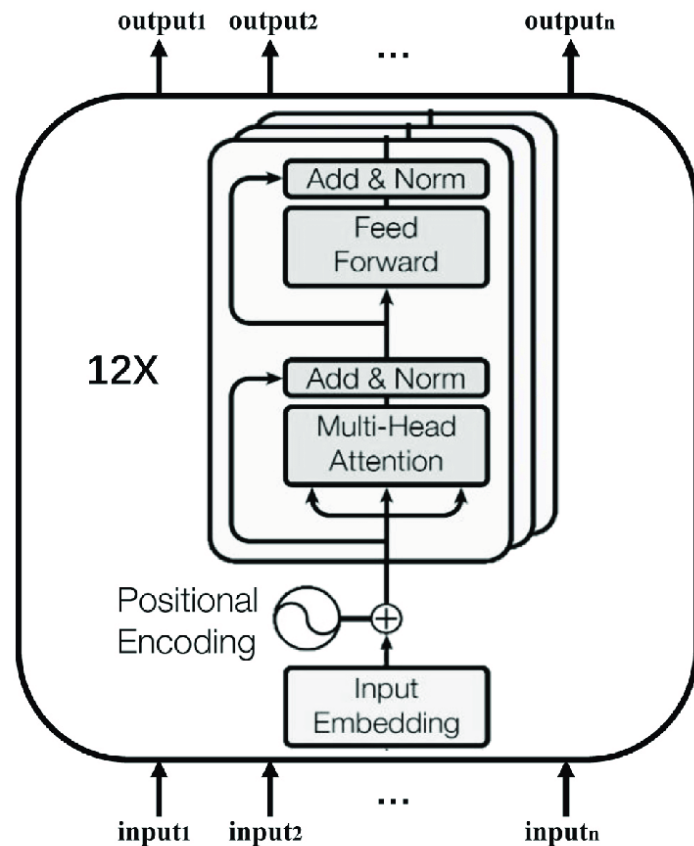
Enhance explainability of Language Models through Knowledge Graphs, while not losing language skills.



# Related Work

## BERT [Devlin et al. (2018)]

- ❑ encoder-only Transformer architecture
  - ❑ trained with Masked Language Modeling (MLM) and Next Sentence Prediction (NSP)
  - ❑ trained on english wikipedia (2500M words) + Toronto BookCorpus (800M words)
- good for NLU/NLI tasks,  
not designed for text generation
- ❑ example use cases:  
Token/Text Classification, Question Answering
  - ❑ BERT<sub>BASE</sub> embedding size: 768



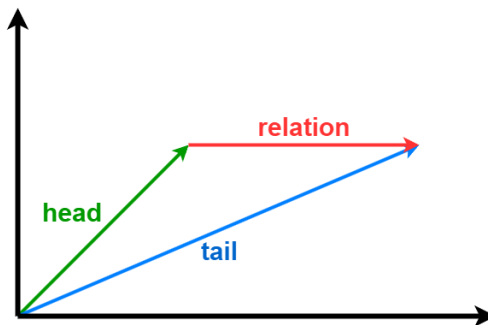
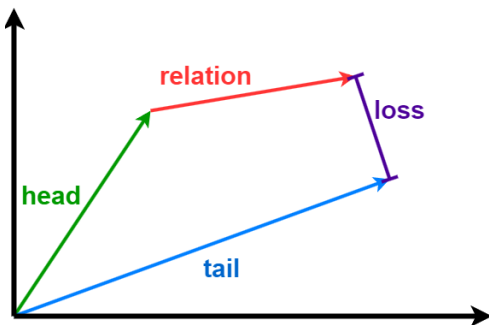
# Related Work

## Knowledge Graph Embedding (KGE)

- ❑ KGs can be embedded in vectorspace → used e.g. for link prediction, clustering
- ❑ embedding can be used for integration into LMs
- ❑ BERT<sub>BASE</sub> has 768-dimensional embedding → use same vectorspace

Translational distance models (e.g. TransE [Bordes et al. (2013)]):

- ❑ every entity (head, tail) and relation gets a vector
  - ❑ vectors should add up (  $head + relation = tail$  )
- distance is our loss (  $loss = \|(head + relation) - tail\|_2$  )

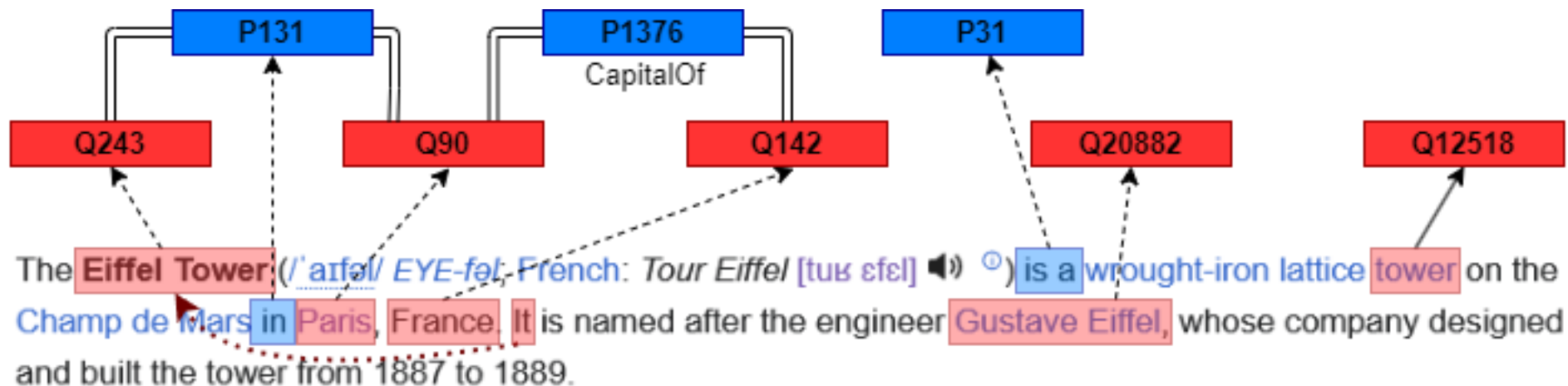


# Related Work

## T-REx Dataset [Elsahar et al. (2017)]

- ❑ dataset of Wikipedia abstracts with Wikidata entities and relations aligned
- ❑ 3.09M Wikipedia abstracts (6.2M sentences)
- ❑ 11M triples 642 unique relations

## Creation Example (non-exhaustive):



# Methods

## Data

- ❑  $\forall$  abstracts: save tokenized text (IDs) + occuring triples in datastructure ("Sample")
- ❑  $\forall$  triples in abstract: save Wikidata ID for head, relation, tail + token boundaries for head and tail in datastructure ("Triple")
- ❑ relations don't necessarily appear in text  $\rightarrow$  use of seperate relation embedding matrix

### Example (simplified):

The Eiffel Tower is a [...] tower [...] in Paris, France.

- ❑ Tokens: [The] [Eiffel] [Tower] [is] [a] [tower] [in] [Paris] [,] [France] [.]
  - ❑ Relation Triples:  
(Eiffel Tower, instance of, tower), (Eiffel Tower, located in, Paris), (Paris, capital of, France)
- $\rightarrow$  in Wikidata IDs: (Q243, P31, Q12518), (Q243, P131, Q90), (Q90, P1376, Q142)
- ❑ token boundaries:  
Eiffel Tower (Q243): [1, 2], tower (Q12518): [5, 5], Paris (Q90): [3, 3], France (Q142): [5, 5]

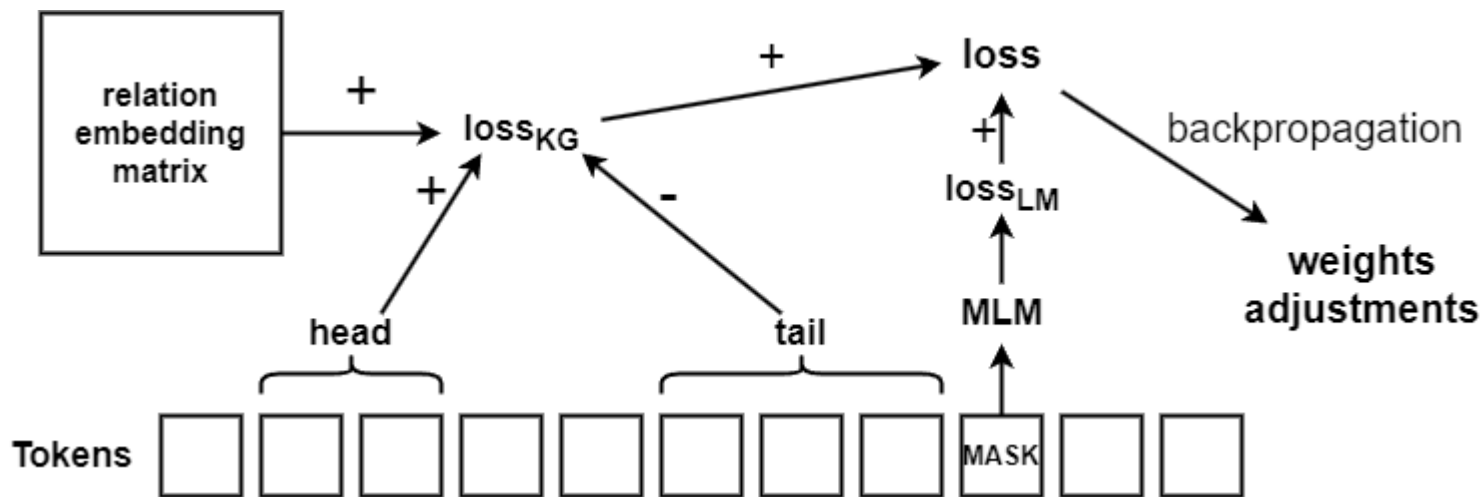


# Methods

## Training

Normal training step for encoder model (MLM) is extended with KG training:

- entity token embeddings are pulled out from LM, averaged
- relation embeddings are taken from embedding matrix (stored separately)
- KG-loss is computed on these embeddings (  $loss = \|(head + relation) - tail\|_2$  )
- LM-loss and KG-loss are combined (  $loss = loss_{LM} + loss_{KG}$  )



# Evaluation

## Language Skills:

- ❑ use of common benchmarks (e.g. GLUE, superGLUE)

## Knowledge Graph:

- ❑ building standalone KGE with same embedding method (e.g. TransE) on same data with different framework
- ❑ evaluate how good the LM-KG-embedding is compared to standalone embedding

# Expected Results

haven't done any tests yet

- ❑ LM-loss didn't get worse → expectation: language skills aren't lost
- ❑ KG-loss looked promising → hopefully LM-KG-embedding is (nearly) as good as standalone embedding

# Future Work

What are the capabilities in knowledge related tasks?

→ knowledge benchmarks (e.g. KILT)

# Conclusion

- ❑ KGs are a structured knowledge bases, can be embedded in vectorspace
- this allows incorporating in LMs
- ❑ use of same vectorspace → use of combined loss for training the LM
- ❑ should increase interpretability of LM embeddings → enhance explainability of LMs
- ❑ should not reduce language skills
- ❑ could enhance results in knowledge tasks