Chapter 9

Word Representations[part2]

World Knowledge

Content of this Chapter

1. Representing World Knowledge: ConceptNet
2. Embedding World Knowledge: ConceptNet Numberbatch
3. Evaluating Word Embeddings
4. Enhanced Representation Through Knowledge Integration

9.1 Representing World Knowledge - ConceptNet

- What is world knowledge?
- Why do we want to use it?
- How to represent it?

World Knowledge

- Humans are very good at understanding language. Why?

- Humans have world knowledge

> World knowledge: In language studies, the non-linguistic information that helps a reader or listener interpret the meanings of words and sentences. Also called extra-linguistic knowledge.
>
> https://www.thoughtco.com/world-knowledge-language-studies-1692508

World Knowledge

- Why do we need world knowledge in NLP?

> *"Romeo and Juliet" is one of Shakespeare's early tragedies. The play has been highly praised by critics for its language and dramatic effect.*

- We know that
  - Shakespeare is a writer

  → tragedy refers to a play rather than an event
  - Romeo and Juliet is a famous play by Shakespeare

  → The play refers to Romeo and Juliet

Example from Ovchinnikova, Ekaterina: *Integration of world knowledge for natural language understanding*. 3 : Springer Science & Business Media, 2012, p.1

World Knowledge

→ World knowledge is important!

- Computers do not possess it

→ Let‘s change this!

Todos:

1. Find a source of world knowledge
2. Model it into word embeddings
3. Show that this improves the embeddings

World Knowledge

- Many resources modelling world knowledge
  - Dbpedia
  - Wikidata
  - BabelNet
  - **ConceptNet**

- We use ConceptNet:
  - Focus on natural language (represents phrases, …)
  - Large vocabulary
  - Provides word embeddings!

ConceptNet

conceptnet.io

- Knowledge Graph
- Built from different sources
  - Open Mind Common Sense (OMCS)
  - Wiktionary
  - Games with a purpose
  - Open Multilingual WordNet
  - …
- Represents word meaning
- Has been used to create word embeddings!

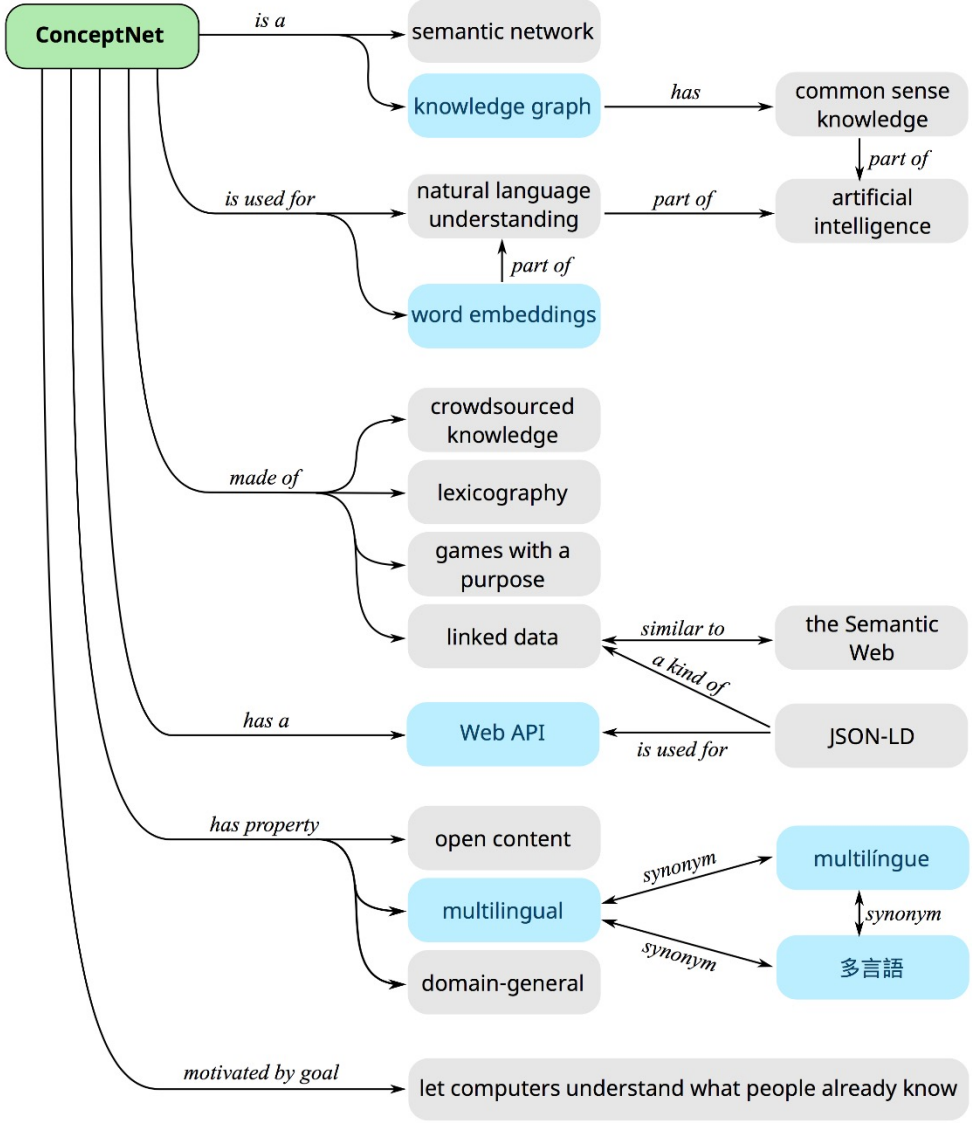
ConceptNet as a Hypergraph

- ConceptNet is a hypergraph $G = (V, E)$
- The nodes V are words or phrases
- The edges E are weighted, directed relations between the nodes
  - Symmetric (Antonym, Synonym, LocatedNear, …) or
  - Asymmetric (AtLocation, Causes, PartOf, UsedFor, …)
- Two nodes are often connected by multiple edges!

- Example: ConceptNet has a RelatedTo-edge from Shakespeare to Hamlet with weight 1.056

ConceptNet

conceptnet.io



Self-representation of ConceptNet

Word Embeddings with World Knowledge

Todos:

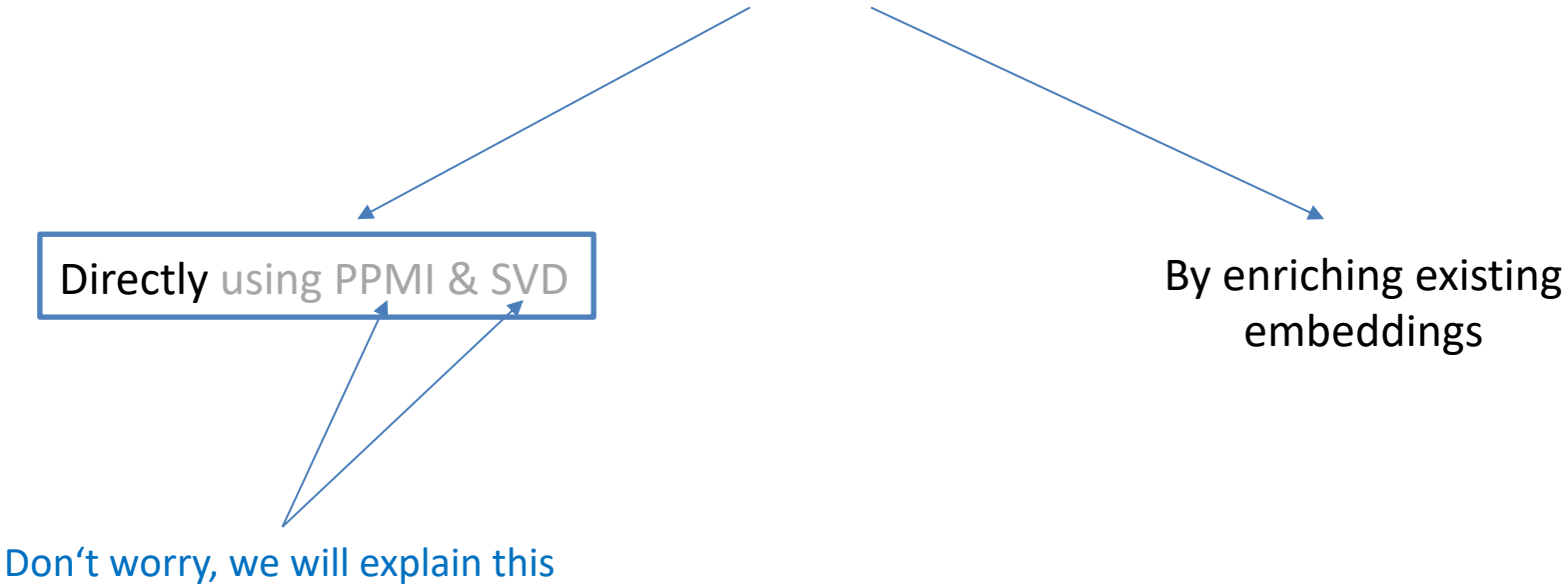
1. Find a source of world knowledge ✅
2. Model it into word embeddings
3. Show that this improves the embeddings

9.2 Embedding World Knowledge – ConceptNet-PPMI & Numberbatch

- How to use world knowledge in word embeddings?

Word Embeddings from ConceptNet

How to get word embeddings from ConceptNet?



Word Embeddings from ConceptNet

- Extract word embeddings from ConceptNet!
- Outline:
  - Convert ConceptNet to a term-term matrix
  - Calculate Positive Pointwise Mutual Information (PPMI) of the matrix entries
  - Reduce the dimensionality by Singular Value Decomposition (SVD)
  - Combine the result into an embedding matrix

Word Embeddings from ConceptNet

- Step 1: Convert ConceptNet to a term-term matrix
  - ConceptNet is a graph $G = (V, E)$
  - Prune the graph: Remove all nodes with less* than three edges (for performance reasons) → N nodes remain, $|V| = N$
  - Create an $N \times N$ matrix M
  - For each cell $m_{i,j}$:
    Set $m_{i,j}$ = sum of weights of all edges between v_i and v_j

Word Embeddings from ConceptNet

- Step 2: Calculate Positive Pointwise Mutual Information (PPMI) of the matrix entries
  - PPMI is a measure of how much two „events“ (here: words/nodes) are related

Pointwise mutual information:

Do events x and y co-occur more than if they were independent?

PMI between two words: (Church & Hanks 1989)

Do words x and y co-occur more than if they were independent?

$$\mathrm{PMI}(word_1, word_2) = \log_2 \frac{P(word_1, word_2)}{P(word_1)P(word_2)}$$

Recap from Text Mining: (P)PMI

- PMI ranges from $-\infty$ to $+\infty$
- But the negative values are problematic
- Things are co-occurring **less than** we expect by chance
  - Unreliable without enormous corpora

- So we just replace negative PMI values by 0
- Positive PMI (PPMI) between $word_1$ and $word_2$:

$$\text{PPMI}(word_1, word_2) = \max\left(\log_2 \frac{P(word_1, word_2)}{P(word_1)P(word_2)}, 0\right)$$

Word Embeddings from ConceptNet

- Step 3: Reduce the dimensionality by Singular Value Decomposition (SVD)
  - SVD splits a matrix M into three components $M = KSD^T$
  - Truncating these components and transforming back to $M' = K'S'(D')^T$ yields a lower-dimensional „approximation“ of M
- SVD covered in Information Retrieval, only basic idea needed here

Embeddings versus Sparse Vectors

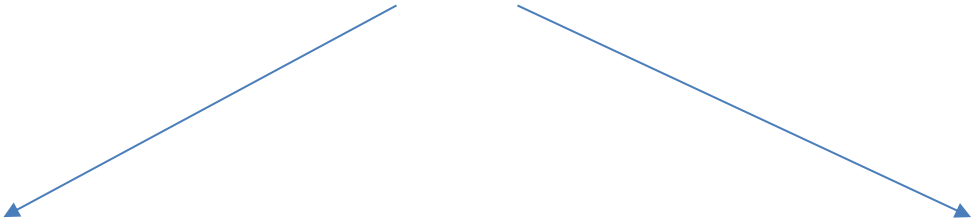
- Dense SVD embeddings sometimes work better than sparse PPMI matrices at tasks like word similarity
  - Denoising: low-order dimensions may represent unimportant information
  - Truncation may help the models generalize better to unseen data
  - Having a smaller number of dimensions may make it easier for classifiers to properly weight the dimensions for the task
  - Dense models may do better at capturing higher order co-occurrence

Word Embeddings from ConceptNet

- Step: Combine the result into an embedding matrix
  - From SVD: $M_s = K_s \times S_s \times (D_s)^T$
  - Create word embeddings from this!
  - We get a set of **word embeddings** and a set of **context embeddings**:
  - Word embeddings: $W^{SVD} = K_s S_s$
  - Context embeddings: $C^{SVD} = (D_s)^T$
  - Combine these by taking the average of the embeddings

Word Embeddings from ConceptNet

How to get word embeddings from ConceptNet?

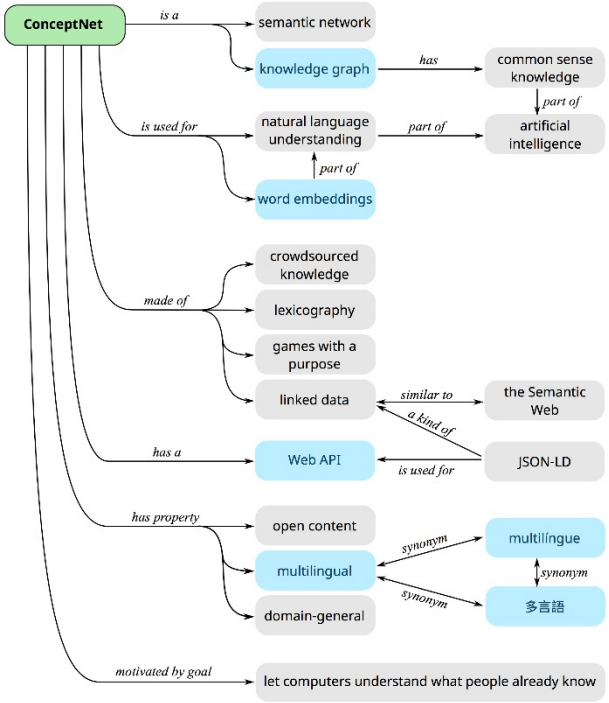


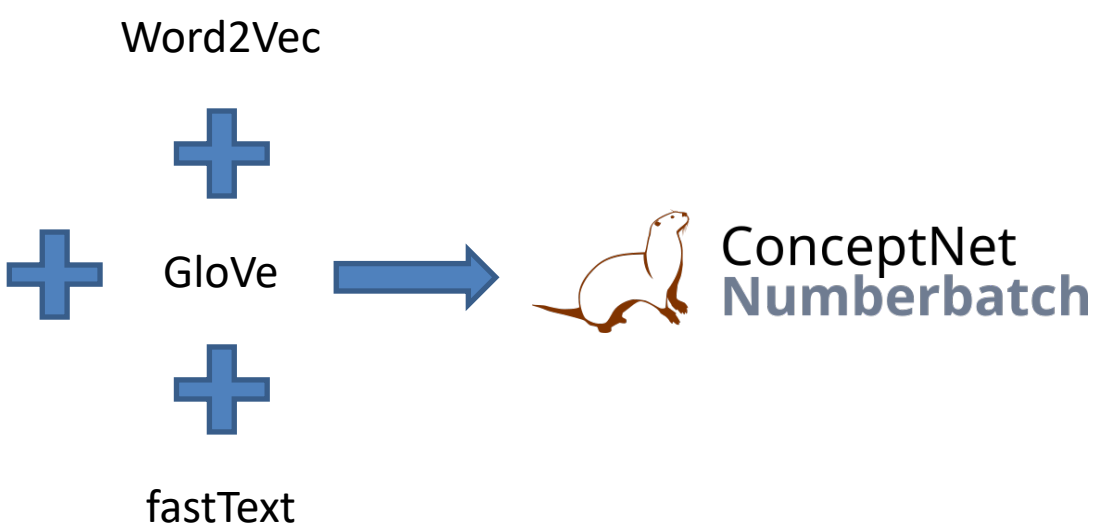
Directly using PPMI & SVD

By enriching existing embeddings

Word Embeddings from ConceptNet

- Idea: Instead of directly creating new embeddings, we can improve existing embeddings!





Word Embeddings from ConceptNet

- How to combine multiple sets of word embeddings with a knowledge graph?

- Two steps:
  - Combine each set of embeddings with the knowledge graph
  - Combine the resulting „enriched“ embeddings with each other

Combining Word Embeddings with a Knowledge Graph

- Use a variant of **Retrofitting**

- Retrofitting:
  - Given a set of unmodified word vectors $\mathrm{u} \in U, |U| = m$ and a graph $\mathrm{G} = (\mathrm{V}, \mathrm{E})$
  - Create a new set of embeddings $\mathrm{w} \in W$ that minimises the following loss function:

$$L(W) = \sum_{i=1}^{m} \left[\alpha_i ||w_i - u_i||^2 + \sum_{(i,j) \in E} \beta_{i,j} ||w_i - w_j||^2 \right]$$

For all words in the vocabulary

Keep w as close to u as possible!
α_i: Weight for the original embedding

Get the word's neighbours in the graph as close as possible!
$\beta_{i,j}$: Edge weights in the graph

Faruqui, Manaal ; Dodge, Jesse ; Jauhar, Sujay K. ; Dyer, Chris ; Hovy, Eduard ; Smith, Noah A.: *Retrofitting Word Vectors to Semantic Lexicons*, 2014. - Proceedings of NAACL 2015

Combining Word Embeddings with a Knowledge Graph

- Speer and Chin:
  - Original optimisation from Faruqui et al. does not perform well for larger corpora
  - Original algorithm does not consider words present in only one vocabulary

→ Propose a new algorithm „Expanded Retrofitting“

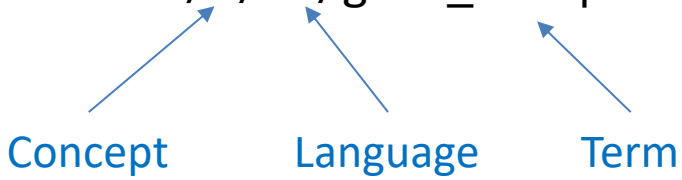
- Steps to this algorithm on the following slides:
  1. Aligning the vocabularies
  2. Normalising the features
  3. Updating the entire matrix at once (opposed to iterative process by Faruqui)

Speer, Robert ; Chin, Joshua: *An Ensemble Method to Produce High-Quality Word Embeddings*, 2016.

Combining Word Embeddings with a Knowledge Graph

- Step 1: Aligning the vocabularies
  - Different pre-trained word embeddings often have different preprocessing steps, like lemmatisation/stemming, stopword removal, …

  → Apply a set of standard transformations (adapted from ConceptNet)
    - „Giving an example“ becomes `/c/en/give_example`



  → Unified vocabulary with rather large overlap
  - Attention: Many-to-one relation!
    Different words mapped to same concept: $\{w_1, w_2, w_3\} \rightarrow c_1$

    →Use the weighted average of the vectors (f is the frequency of the word)

    →$e(c_1) = f(w_1)e(w_1) + f(w_2)e(w_2) + f(w_3)e(w_3)$

Speer, Robert ; Chin, Joshua: *An Ensemble Method to Produce High-Quality Word Embeddings*, 2016.

Combining Word Embeddings with a Knowledge Graph

- Step 2: Normalising the features
  - Following Pennington et al. (GloVe):
    Normalise the columns (i.e. the embedding vectors) of the matrix using L2-norm
  - Speer and Chin find that L1-norm works even better

  → Intuition:
  Increase weight of distinguishing features, reduce weight of noisy features

Speer, Robert ; Chin, Joshua: *An Ensemble Method to Produce High-Quality Word Embeddings*, 2016.

Combining Word Embeddings with a Knowledge Graph

- Step 3: Updating the entire matrix at once
  - Assume the following variables:
    - U: the original embedding
    - m': the size of the merged vocabulary (resulting from step 1)
    - n: the size of embeddings in U
    - W^0: $m' \times n$ matrix holding the initial embeddings in its rows. Words not covered by U are zero-vectors.
    - A: diagonal matrix. $A_{i,i} = 1$ if $w_i \in U$, 0 otherwise. Used to keep words in U close to their original embeddings, while allowing words not in U to change
  - We will describe an update step reducing the already defined loss!

Recall: $L(W) = \sum_{i=1}^{m} \left[\alpha_i ||w_i - u_i||^2 + \sum_{(i,j) \in E} \beta_{i,j} ||w_i - w_j||^2 \right]$

Speer, Robert ; Chin, Joshua: *An Ensemble Method to Produce High-Quality Word Embeddings*, 2016.

Combining Word Embeddings with a Knowledge Graph

U: original embedding
m': size of the merged vocabulary
n: size of embeddings in U
W^0: $m' \times n$ matrix, initial embeddings
A: diagonal matrix. $A_{i,i} = 1$ iff. $w_i \in U$

- Step 3: Updating the entire matrix at once
  - Transform ConceptNet into a matrix S by this recipe:
    - ConceptNet contains multiple relations between words w_i and w_j
      → Sum up their edge weights to get $S'_{i,j}$
    - $S'_{i,j}$: association of words w_i and w_j
    - Empirically helpful: Adding „self-edges“
      → $S'_{i,i} = S'_{i,i} + 1$
    - Apply L1-norm to the rows of the resulting matrix S'
      → S

Speer, Robert ; Chin, Joshua: *An Ensemble Method to Produce High-Quality Word Embeddings*, 2016.

Combining Word Embeddings with a Knowledge Graph

U: original embedding
m': size of the merged vocabulary
n: size of embeddings in U
W^0: $m' \times n$ matrix, initial embeddings
A: diagonal matrix. $A_{i,i} = 1$ iff. $w_i \in U$
S: association matrix from ConceptNet

- Step 3: Updating the entire matrix at once
  - Update step:

$$W^{k+1} = normalize\left(\left[\left(SW^k + AW^0\right)^T (I + A)^{-1}\right]^T\right)$$

Get the vectors „closer“ to the association matrix

Keep the vectors close to the original embedding, if they are in it

Normalisation: Words in U appear twice in the first part of the formula. This effectively halves their values

Speer, Robert ; Chin, Joshua: *An Ensemble Method to Produce High-Quality Word Embeddings*, 2016.

Word Embeddings from ConceptNet

- How to combine multiple sets of word embeddings with a knowledge graph?

- Two steps:
  - Combine each set of embeddings with the knowledge graph ✅
  - **Combine the resulting „enriched“ embeddings with each other**

Word Embeddings from ConceptNet

- By retrofitting, we get vectors for
  - Word2Vec+ConceptNet
  - GloVe+ConceptNet
  - fastText+ConceptNet
- Combine these!

→ Concatenate all embeddings

→ Perform truncated SVD (as for ConceptNet-PPMI) to get back to 300 dimensions!

Word Embeddings with World Knowledge

Todos:

1. Find a source of world knowledge ✅
2. Model it into word embeddings ✅
3. Show that this improves the embeddings

9.3 Evaluating Word Embeddings

- How to assess the quality of word embeddings?
- How to compare two word embeddings?

Evaluating Word Embeddings

Evaluating word embeddings

Extrinsic

Evaluating the performance by some NLP-task

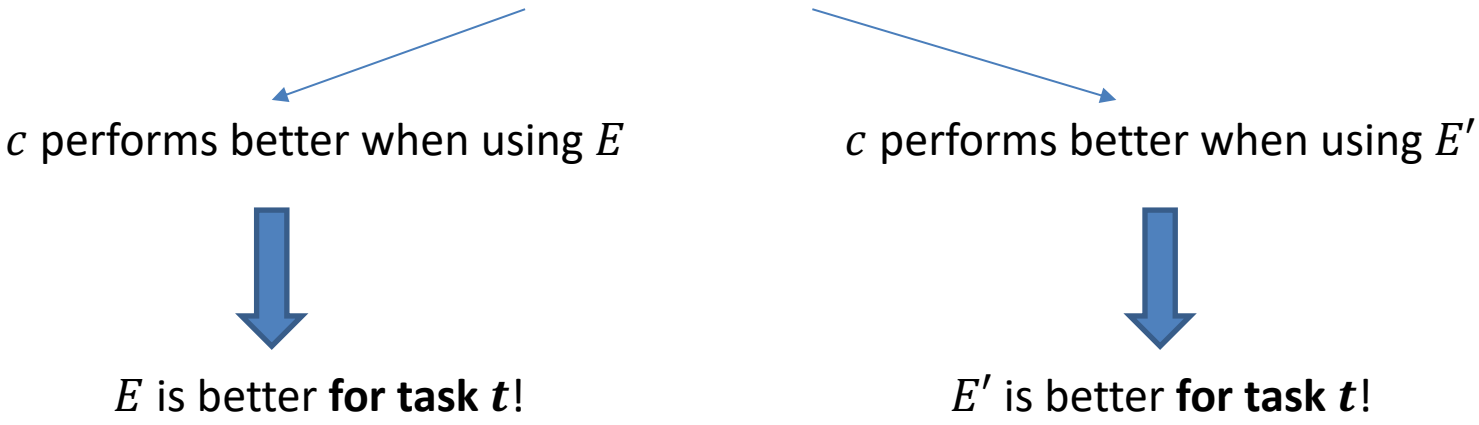
Intrinsic

Testing the ability to model word similarity

Evaluating Word Embeddings

- Extrinsic evaluation:
  - Given two sets of embeddings E and E'
  - Select some NLP-task t…
  - … and a classifier c for t that makes use of embeddings

  - Evaluate the performance of classifier c on task t using each set of embeddings



Evaluating Word Embeddings

- Intrinsic evaluation:
  - Need some kind of gold standard to compare to!

  → **Human Intuition Datasets (HID)**
    - Datasets of word pairs annotated by humans with their (relative) relatedness r

    → r(table, chair) > r(duck, plane)

Evaluating Word Embeddings – HIDs

- HIDs rate word pairs by their relatedness
- Many such datasets exist
- Usually created by crowdsourcing or expert annotations
- Examples:
  - WS-353
  - MEN-3000
  - Stanford Rare Words (RW)

Evaluating Word Embeddings – HIDs

- WordSimilarity-353 (WS-353)
  - Finkelstein et al., 2001
  - Dataset of 353 word pairs with human similarity judgements
  - Rates the similarity on a scale from 0 to 10
  - E.g.:
    - r(tiger, cat) = 6.77
    - r(stock, egg) = 1.81
  - Note: Only 353 word pairs! But still standard evaluation dataset

Evaluating Word Embeddings – HIDs

- Stanford Rare Words (RW)
  - Luong et al., 2013
  - Focus on rare words (underrepresented in most datasets!)
  - 2034 word pairs
  - Similiarity scores from 0 to 10
  - E.g.:
    - r(yodelling, singing) = 7.50
    - r(ruralist, advocate) = 0.67

Evaluating Word Embeddings – HIDs

- MEN-3000
  - Bruni et al., 2014
  - Different approach: Given two pairs of words (a&b and c&d), is a more strongly related to b than c is to d?
  - Each pair is rated against 50 comparison pairs
  - Final rating for pair p: how many times is p rated higher than the other pair?
  - Annotation done by crowdsourcing
  - E.g.:
    - r(sun, sunlight) = 50
    - r(restaurant, violet) = 7

Evaluating Word Embeddings on HIDs

- Evaluation done by Speer and Chin
- Spearman rank correlation between the embeddings and HIDs

Dataset\Embedding	Word2Vec	GloVe	Conceptnet Numberbatch
MEN-3000	0.732	0.787	**0.866**
Stanford RW	0.374	0.148	**0.601**
WS-353	0.622	0.676	**0.828**

Word Embeddings with World Knowledge

Todos:

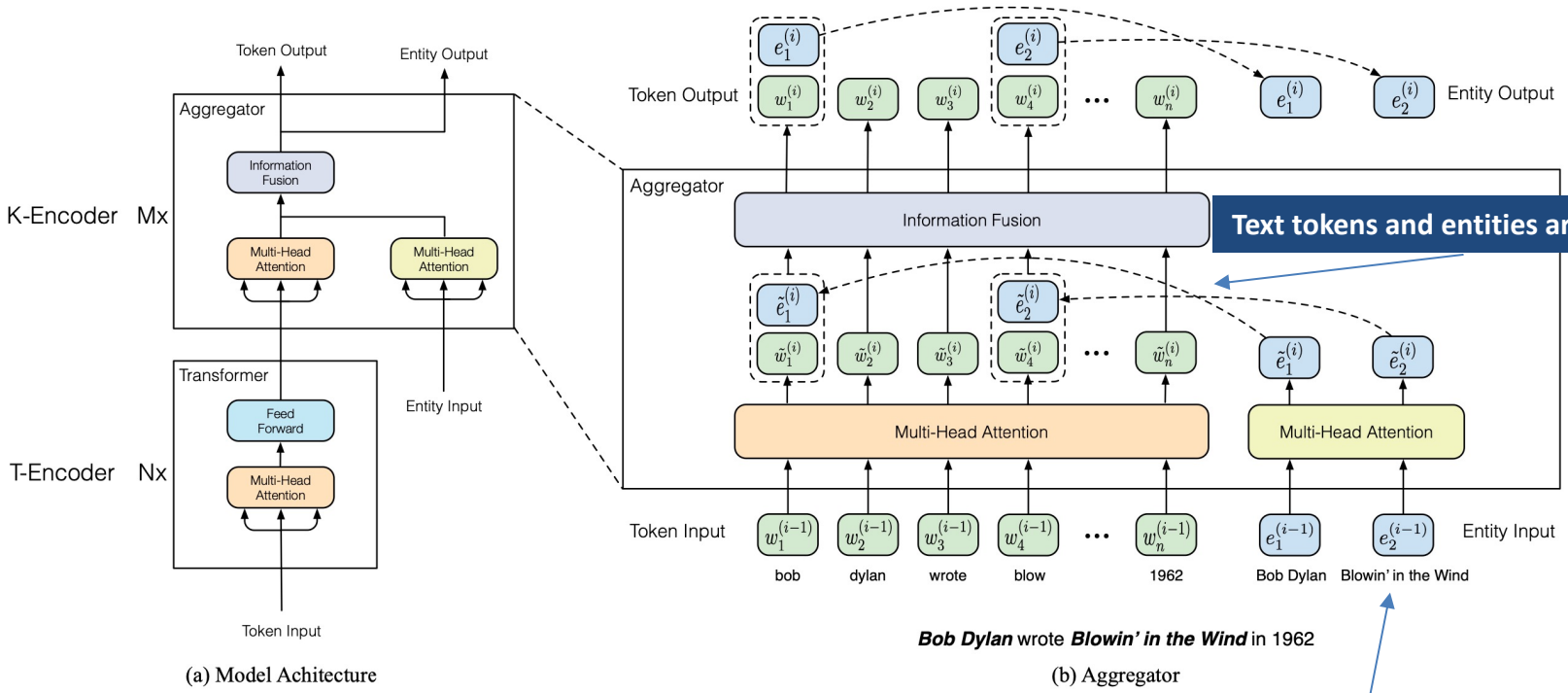
1. Find a source of world knowledge ✅
2. Model it into word embeddings ✅
3. Show that this improves the embeddings ✅

9.4a Integrating World Knowledge into Language Models

Using knowledge graphs to make better language models
→ better contextualized word embeddings

→ Meet BERTS's friend ERNIE

ERNIE — Architecture



T-Encoder captures underlying structures from the **text** (“Bob Dylan” are just two words).
K-Encoder integrates token-oriented **knowledge** by specifically learning entity embeddings (“Bob Dylan” represents the musician and writer).

Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., & Liu, Q. (2019, July). ERNIE: Enhanced Language Representation with Informative Entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 1441-1451).

ERNIE — Training

- Dataset: Wikipedia text corpus
- Align text to the entities from WikiData (KG based on Wikipedia)
- Training tasks:
  - BERT's **Masked Language Model (MLM)** and **Next Sentence Prediction (NSP)** tasks
  - **New task for Knowledge Injection:**
    1. "In 5% of the time, for a given token-entity alignment, we replace the entity with another random entity, *which aims to train our model to correct the errors that the token is aligned with a wrong entity*;
    2. In 15% of the time, we mask token-entity alignments, *which aims to train our model to correct the errors that the entity alignment system does not extract all existing alignments*;
    3. In the rest of the time, we keep token-entity alignments unchanged, *which aims to encourage our model to integrate the entity information into token representations for better language understanding*."

Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., & Liu, Q. (2019, July). ERNIE: Enhanced Language Representation with Informative Entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 1441-1451).

ERNIE — Results

1. ERNIE shows SotA performance on the Entity Typing and Relation Classification tasks

"Given an entity mention and its context, **entity typing** requires systems to label the entity mention with its respective semantic types."

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 (%).

"**Relation classification** aims to determine the correct relation between two entities in a given sentence"

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 (%).

Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., & Liu, Q. (2019, July). ERNIE: Enhanced Language Representation with Informative Entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 1441-1451).

ERNIE — Results

2. ERNIE does not show large performance differences to BERT on other NLP tasks
→ Adding World Knowledge only improves the embeddings

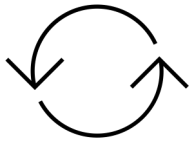
Model	MNLI-(m/mm) 392k	QQP 363k	QNLI 104k	SST-2 67k
$BERT_{BASE}$	84.6/83.4	71.2	-	93.5
ERNIE	84.0/83.2	71.2	91.3	93.5
Model	**CoLA 8.5k**	**STS-B 5.7k**	**MRPC 3.5k**	**RTE 2.5k**
$BERT_{BASE}$	52.1	85.8	88.9	66.4
ERNIE	52.3	83.2	88.2	68.8

Table 6: Results of BERT and ERNIE on different tasks of GLUE (%).

Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., & Liu, Q. (2019, July). ERNIE: Enhanced Language Representation with Informative Entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 1441-1451).

9.4b Integrating World Knowledge into Language Models

Using the knowledge captured in language models for creating better knowledge graphs



Using knowledge graphs to make better language models

Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

data science

LM4KG: Improving Common Sense Knowledge Graphs with Language Models

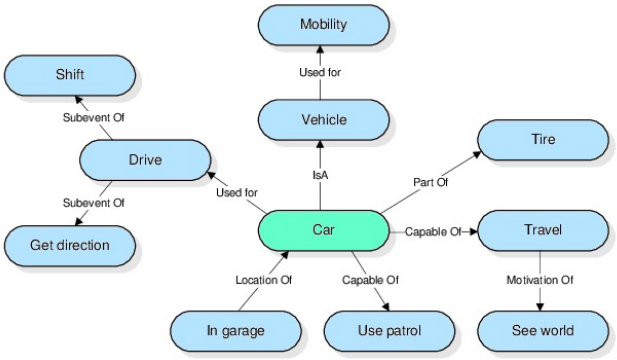
Janna Omeliyanenko, Albin Zehe, Lena Hettinger, Andreas Hotho

Agenda

- Motivation
  - Knowledge Graphs & Language Models
  - Problem Definition
- REWEIGHT
  - Deriving Edge Weights from Language Models
- Evaluation
  - Results: Improving Word Embeddings for Semantic Relatedness
  - Observations
- Conclusion

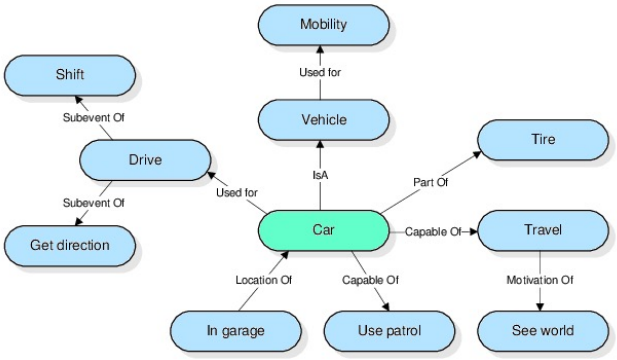
Motivation

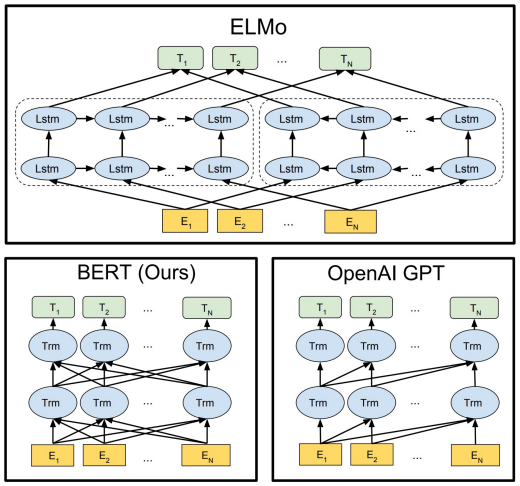
- Knowledge Graphs (**KGs**)
  - Semantic Web
  - Structured
  - Manual/rule-based generation



Motivation

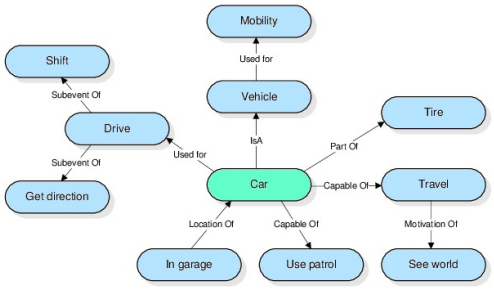
- Knowledge Graphs (**KGs**)
  - Semantic Web
  - Structured
  - Manual/rule-based generation
- Language Models (**LMs**)
  - Machine Learning
  - Unstructured
  - Statistics from large masses of text

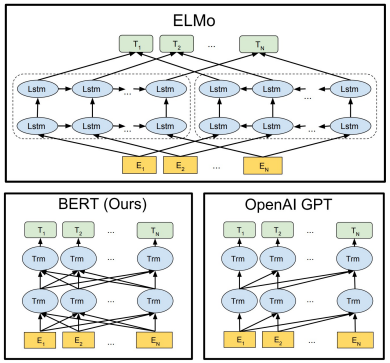




BERT Devlin, Jacob, et al. 2018 arXiv

Motivation





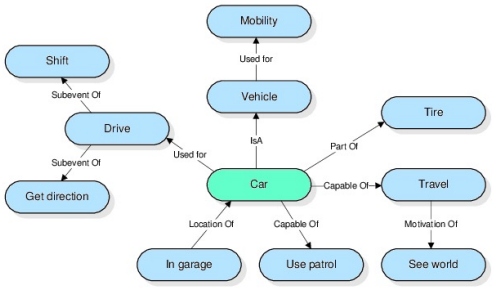
KnowBert
Peters 2019
EMNLP-IJCNLP

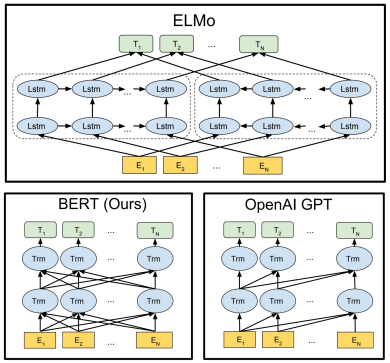
ERNIE
Zhang 2019
arXiv

K-BERT
Liu 2019 AAAI

Stronger LMs

Motivation





KnowBert
Peters 2019
EMNLP-IJCNLP

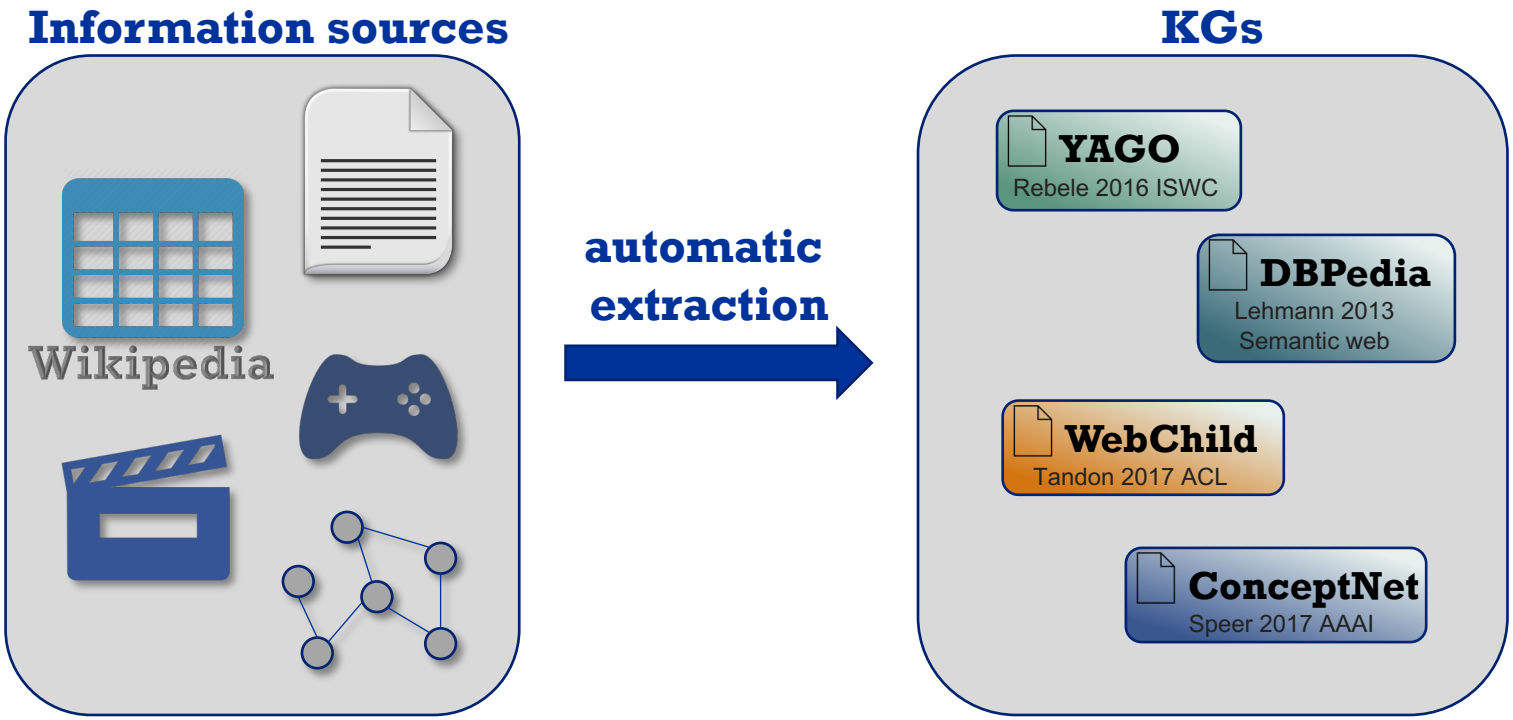
ERNIE
Zhang 2019
arXiv

K-BERT
Liu 2019 AAAI

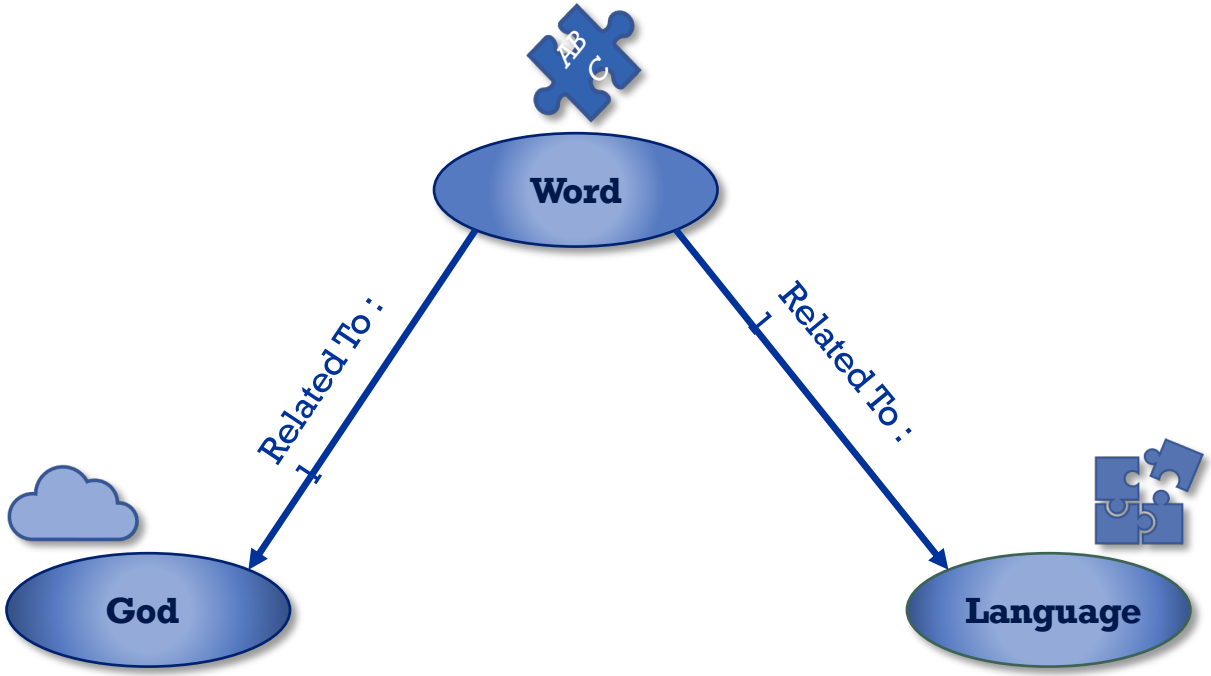
Stronger KGs?

Stronger LMs

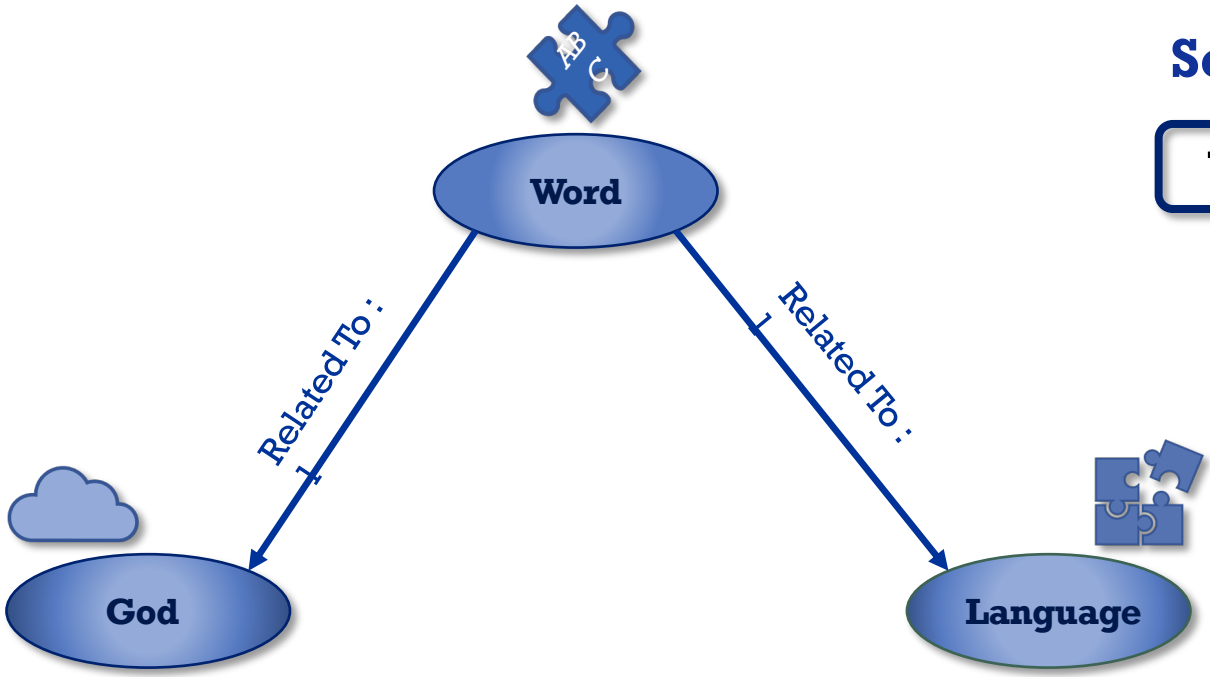
Automatic KGs



Current KGs



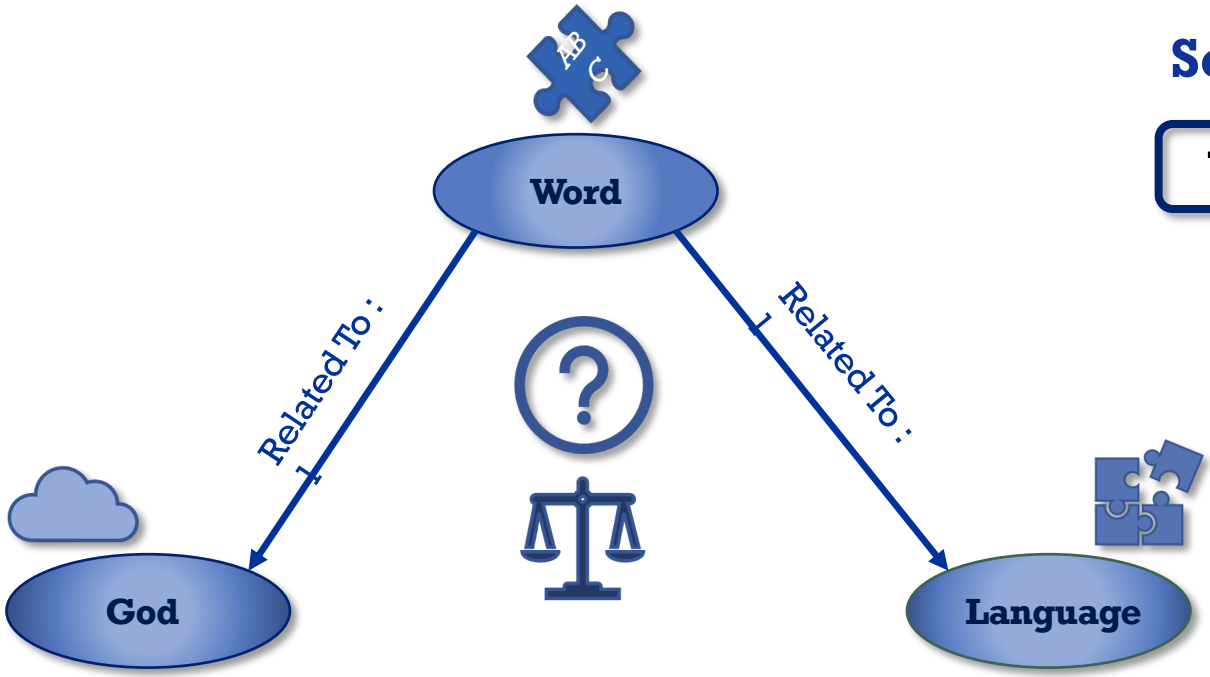
Current KGs



Search:

Word

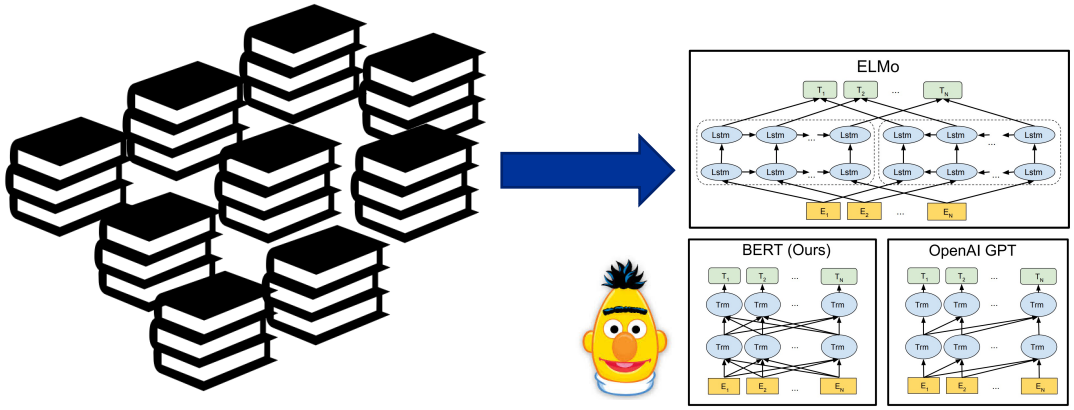
Current KGs



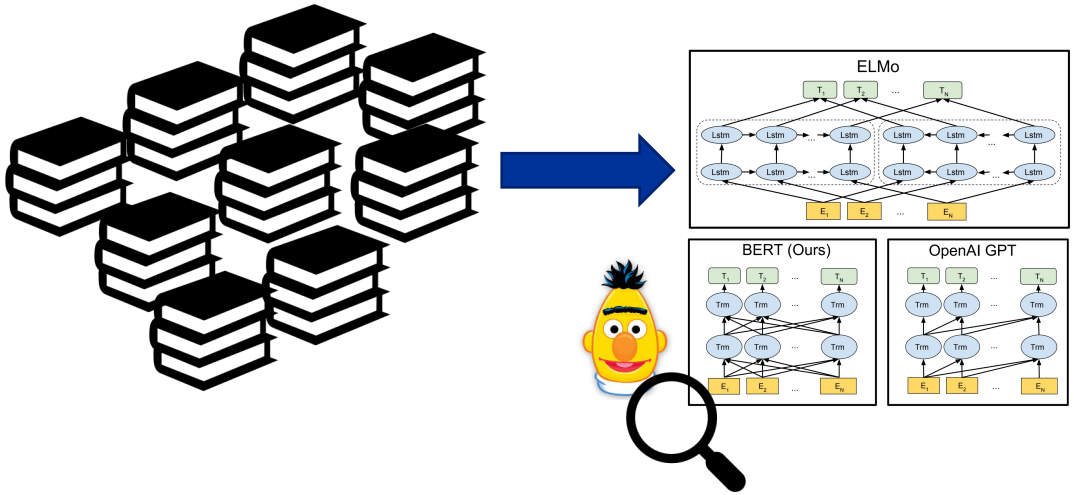
Search:

Word

LMs for KGs



LMs for KGs



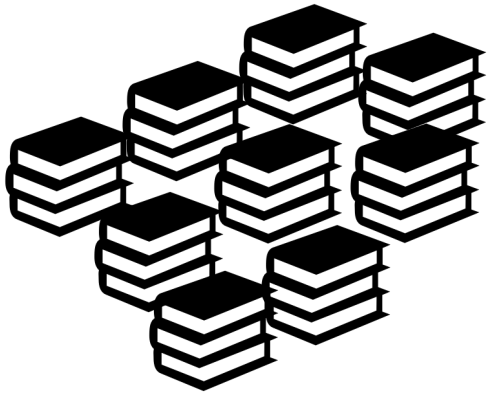
Common Sense

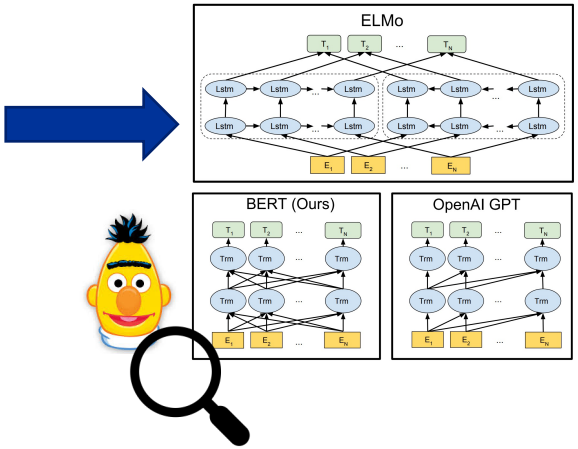
LMs as KBs
Petroni 2019 EMNLP

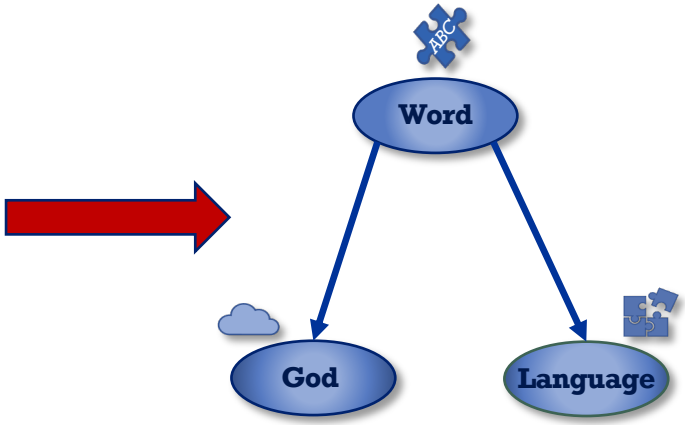
LM Multitask
Radford 2019 OpenAI

BERT Knowledge
Bouraoui et al. 2019 AAAI

LMs for KGs







Common Sense

LMs as KBs
Petroni 2019 EMNLP

LM Multitask
Radford 2019 OpenAI

BERT Knowledge
Bouraoui et al. 2019 AAAI

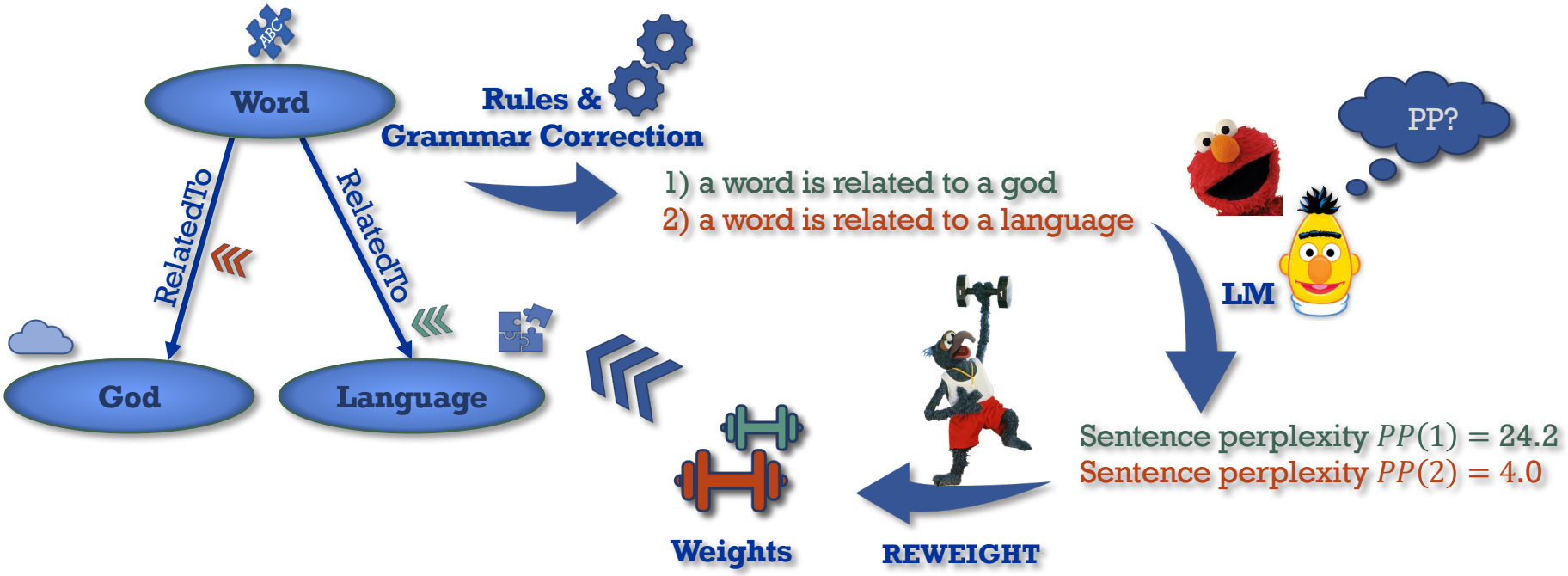
Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

data science

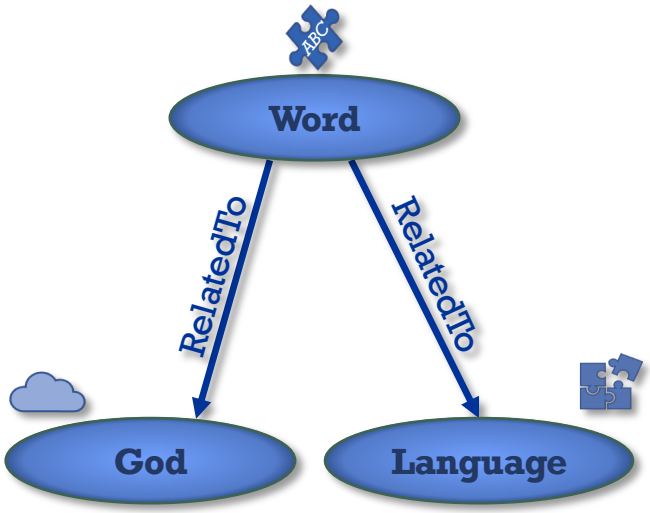
REWEIGHT

Deriving Edge Weights from Language Models

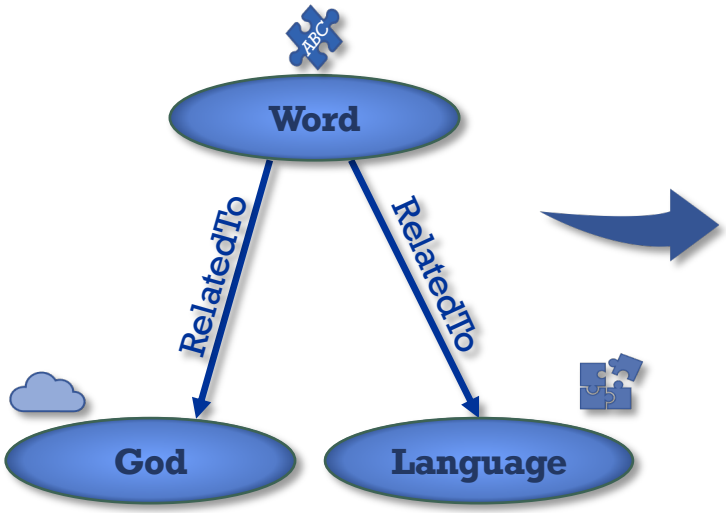
REWEIGHT


Word
RelatedTo
RelatedTo
God
Language
Rules &
Grammar Correction
1) a word is related to a god
2) a word is related to a language
PP?
LM
Sentence perplexity $PP(1) = 24.2$
Sentence perplexity $PP(2) = 4.0$
Weights
REWEIGHT

REWEIGHT

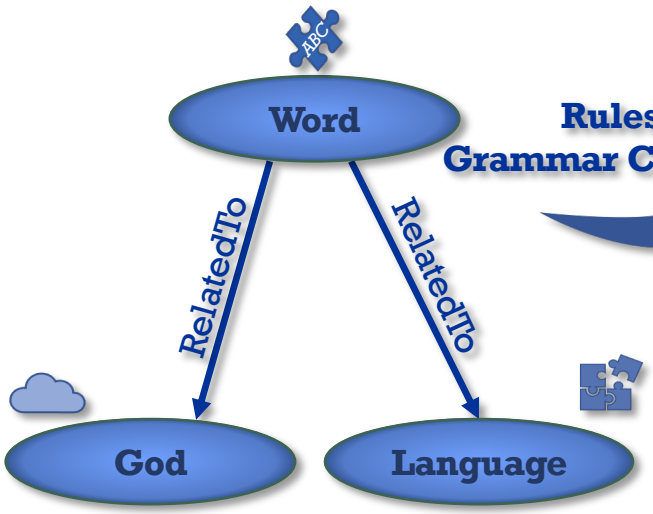


REWEIGHT



1) a word is related to a god
2) a word is related to a language

I: Tripel to Sentence

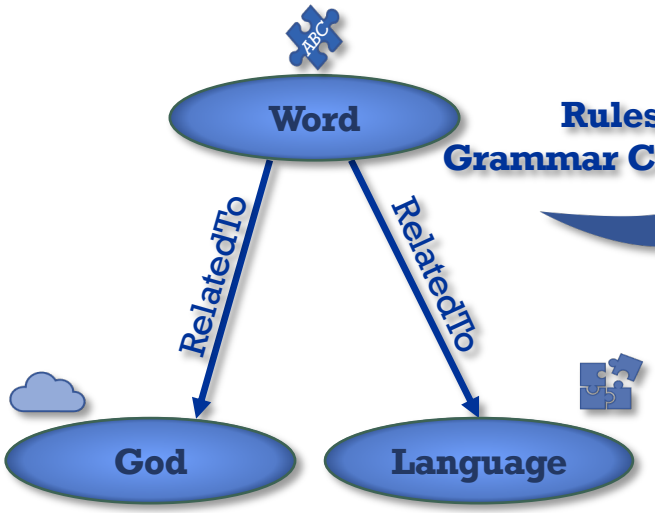


Rules & Grammar Correction

1) a word is related to a god
2) a word is related to a language

Relation	Inverted	Mapping	Counts
/r/AtLocation		is located at	20913
/r/CapableOf		is able to	5398
/r/Causes	*	is caused by	7276
/r/CausesDesire		makes to want to	2028
/r/CreatedBy		created by	195
/r/DefinedAs		is defined as	128
/r/DerivedFrom		is derived from	240851
/r/Desires		want	1051
/r/EtymologicallyDerivedFrom		is derived from	39
...		...	
...			

I: Tripel to Sentence



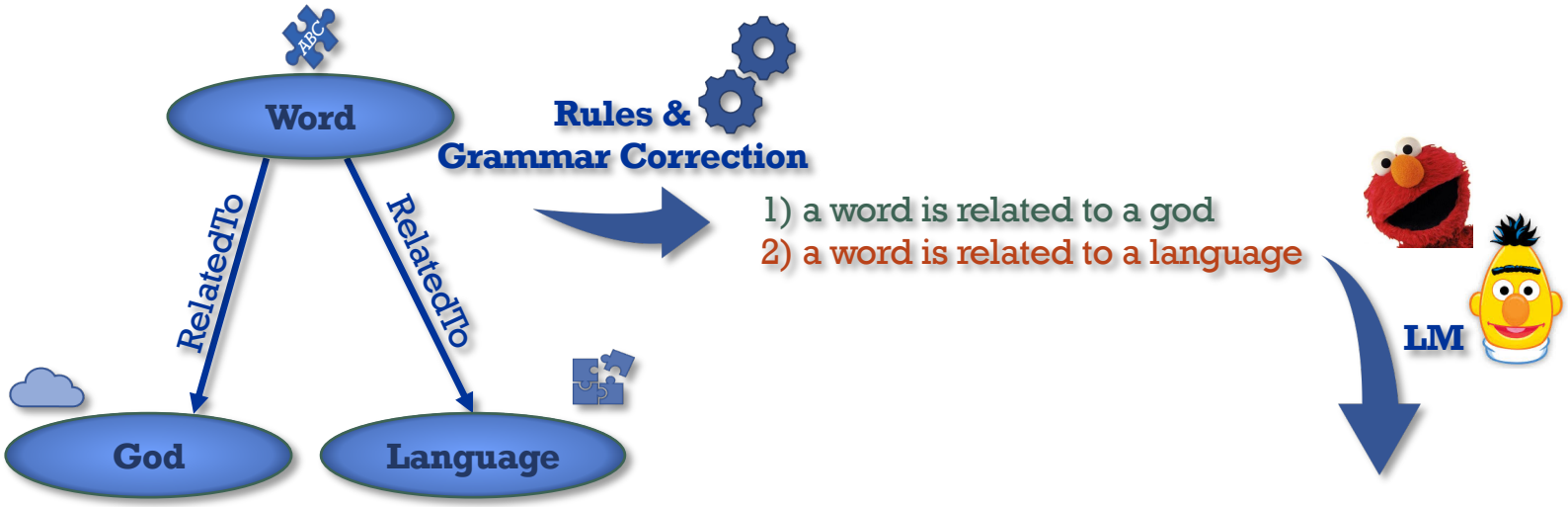
Rules & Grammar Correction

Relation		Inverted Mapping	Counts
/r/AtLocation		is located at	20913
/r/CapableOf		is able to	5398
/r/Causes	*	is caused by	7276
/r/CausesDesire		makes to want to	2028
/r/CreatedBy		created by	195
/r/DefinedAs		is defined as	128
/r/DerivedFrom		is derived from	240851
/r/Desires		want	1051
/r/EtymologicallyDerivedFrom		is derived from	39
...		...	
...			

1) a word is related to a god
2) a word is related to a language

- Grammar checker LM PIE

REWEIGHT


Word
RelatedTo
RelatedTo
God
Language
Rules &
Grammar Correction
1) a word is related to a god
2) a word is related to a language
LM

II: Sentence Rating

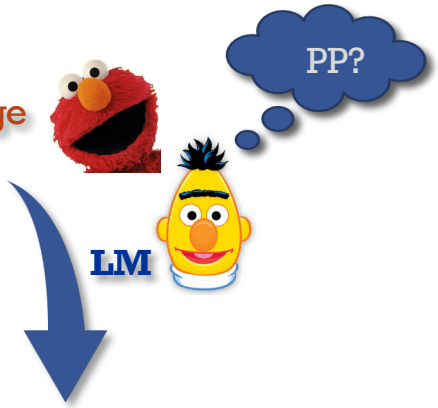
1) a word is related to a god
2) a word is related to a language

- Hypothesis:
  - LMs contain world knowledge
  - LMs likely generate reasonable sentences

  →Sentence likelihood as measure for reasonability

II: Sentence Rating

1) a word is related to a god
2) a word is related to a language



- Hypothesis:
  - LMs contain world knowledge
  - LMs likely generate reasonable sentences

  →Sentence likelihood as measure for reasonability

- Likelihood of a sentence: „Perplexity“ (PP) in bi-directional LMs

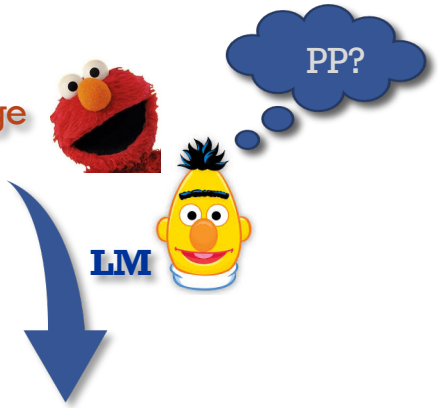
NNLM
Chen 2017
IEEE

$$PP(W) = e^{\frac{1}{N} - \log P(W)},$$

$$P(W) \approx \prod_{i=1}^{N} p(w_i | w_1, \ldots, w_{i-1}, w_{i+1}, \ldots, w_N)$$

II: Sentence Rating

1) a word is related to a god
2) a word is related to a language



Sentence perplexity $PP(1) = 24.2$
Sentence perplexity $PP(2) = 4.0$

$$PP(W) = e^{\frac{1}{N} - \log P(W)},$$

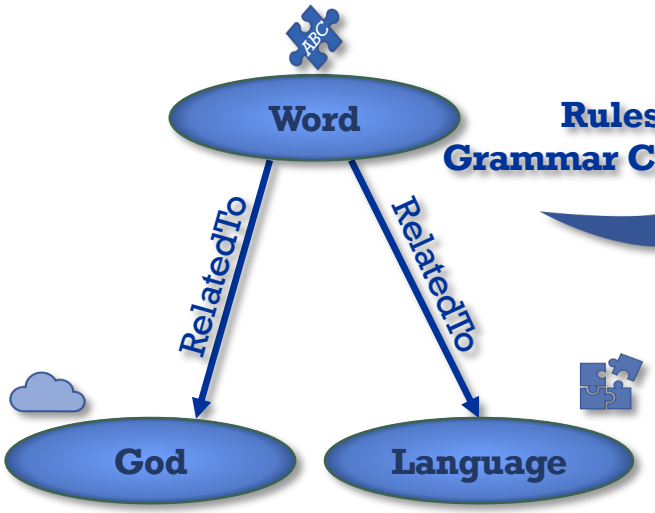
$$P(W) \approx \prod_{i=1}^{N} p(w_i | w_1, \ldots, w_{i-1}, w_{i+1}, \ldots, w_N)$$

- Hypothesis:
  - LMs contain world knowledge
  - LMs likely generate reasonable sentences

  →Sentence likelihood as measure for reasonability

- Likelihood of a sentence: „Perplexity“ (PP)

REWEIGHT



Rules & Grammar Correction

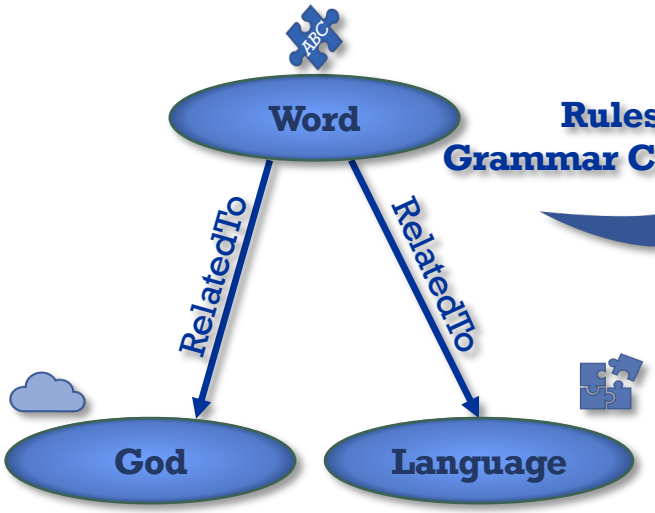
1) a word is related to a god
2) a word is related to a language

LM

Sentence perplexity $PP(1) = 24.2$
Sentence perplexity $PP(2) = 4.0$

Weights

REWEIGHT



Rules & Grammar Correction

1) a word is related to a god
2) a word is related to a language

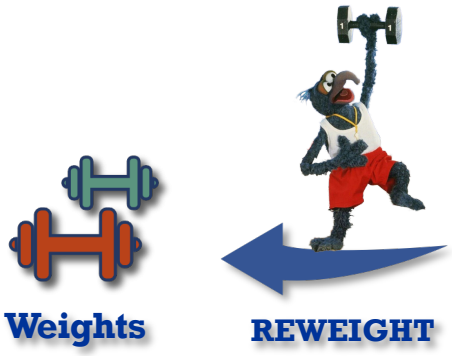
LM

Sentence perplexity $PP(1) = 24.2$
Sentence perplexity $PP(2) = 4.0$

REWEIGHT

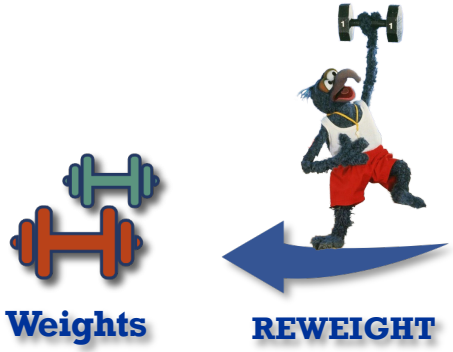
Weights

III: REWEIGHTing



Sentence perplexity $PP(1) = 24.2$
Sentence perplexity $PP(2) = 4.0$

III: REWEIGHTing



Sentence perplexity $PP(1) = 24.2$
Sentence perplexity $PP(2) = 4.0$

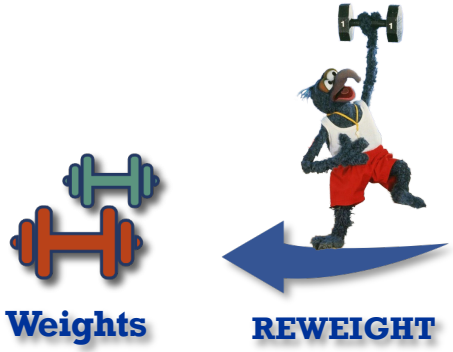
$REWEIGHT_{light}$

perplexity (PP)

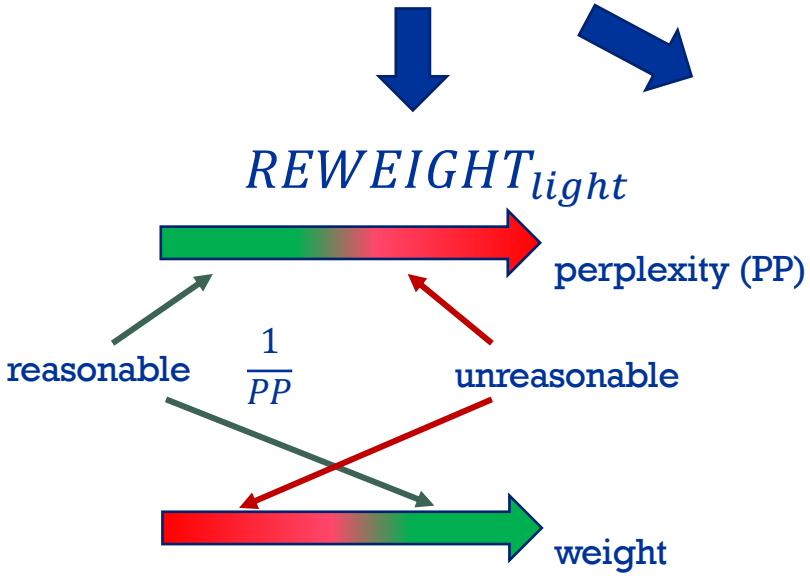
reasonable

unreasonable

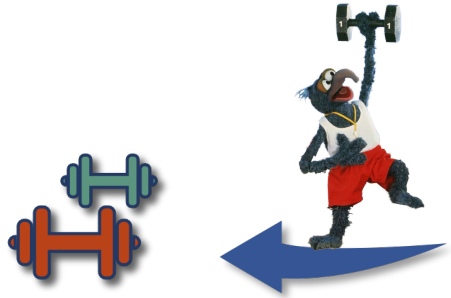
III: REWEIGHTing



Sentence perplexity $PP(1) = 24.2$
Sentence perplexity $PP(2) = 4.0$

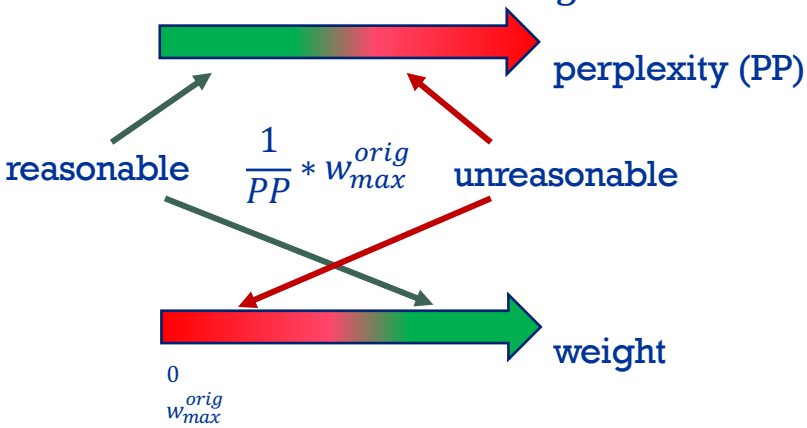


III: REWEIGHTing



Sentence perplexity $PP(1) = 24.2$
Sentence perplexity $PP(2) = 4.0$

$REWEIGHT_{light}$



III: REWEIGHTing

Weights

REWEIGHT

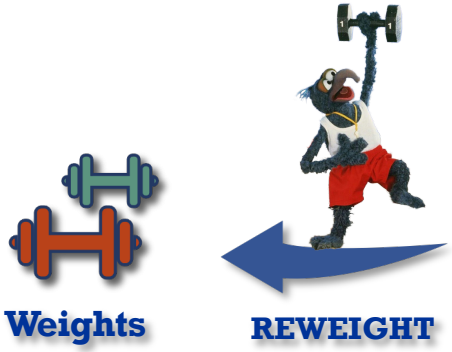
Sentence perplexity $PP(1) = 24.2$
Sentence perplexity $PP(2) = 4.0$

$REWEIGHT_{light}$

$REWEIGHT_{mod}$

perplexity (PP)

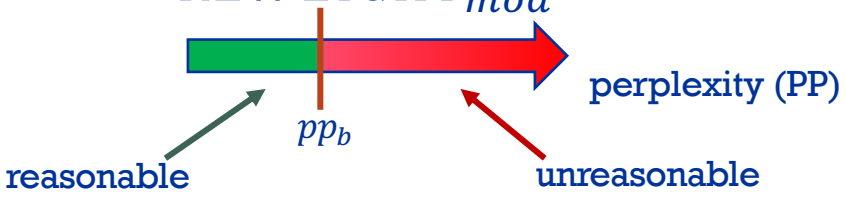
III: REWEIGHTing



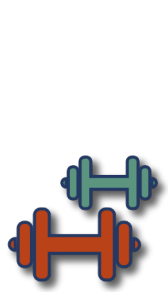
Sentence perplexity $PP(1) = 24.2$
Sentence perplexity $PP(2) = 4.0$

$REWEIGHT_{light}$

$REWEIGHT_{mod}$



III: REWEIGHTing



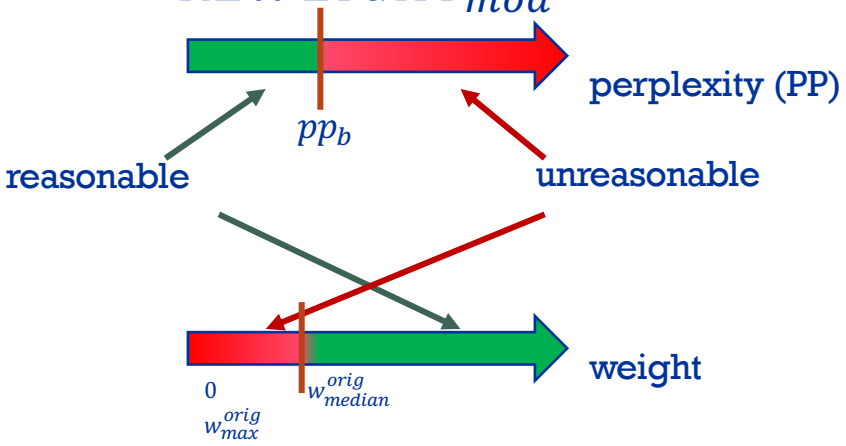
Weights

REWEIGHT

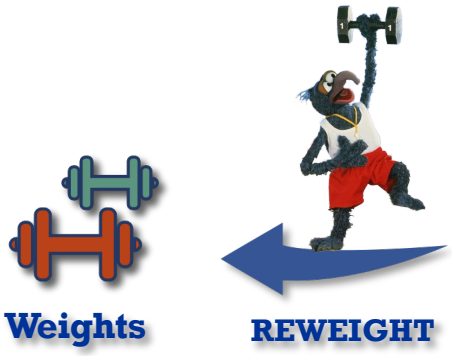
Sentence perplexity $PP(1) = 24.2$
Sentence perplexity $PP(2) = 4.0$

$REWEIGHT_{light}$

$REWEIGHT_{mod}$



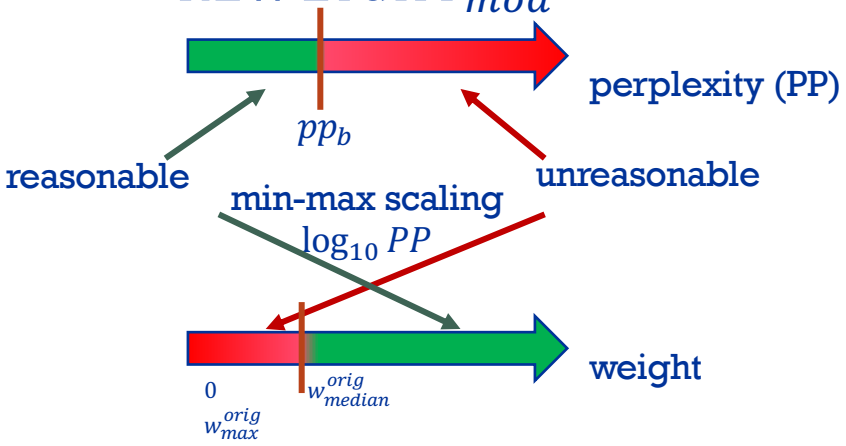
III: REWEIGHTing



Sentence perplexity $PP(1) = 24.2$
Sentence perplexity $PP(2) = 4.0$

$REWEIGHT_{light}$

$REWEIGHT_{mod}$



REWEIGHTing Formulas

$REWEIGHT_{light}$

$$x = \beta_{RWL}(e) := \frac{\gamma_{max}}{pp(e)} ,$$

γ_{max} is the maximum weight in the original KG

$REWEIGHT_{mod}$

Reasonable perplexity border

$$\beta_{RWM}(e) := \begin{cases} r(e), if\ pp(e) < pp_b \\ u(e), otherwise \end{cases}$$

Inverting

$$pp^{inv}(e) = \log_{10}\left(\frac{pp_{max}}{pp(e)}\right)$$

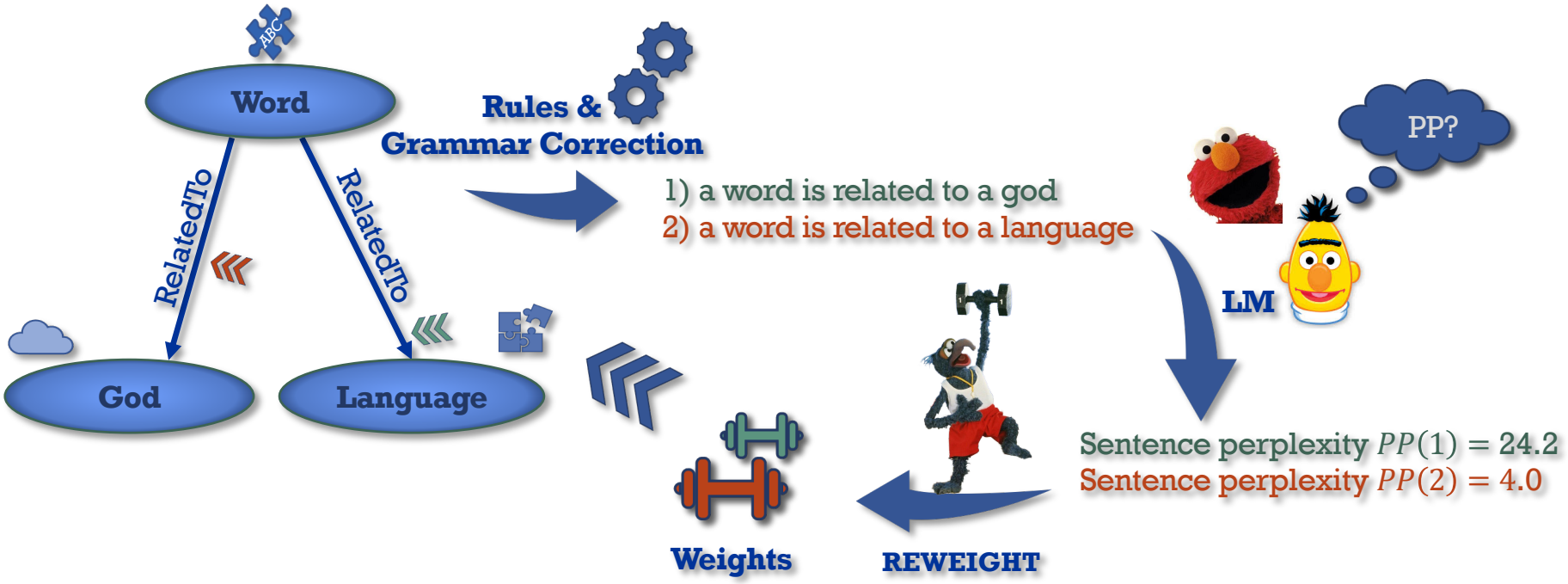
mapping unreasonable sentences

$$u(e) := \frac{\tilde{\gamma}\ pp^{inv}(e)}{pp_b^{inv}}, \quad p_b^{inv} := \log_{10}(\frac{pp_{max}}{pp_b})$$

mapping reasonable sentences

$$r(e) := \tilde{\gamma} + (\gamma_{max} - \tilde{\gamma}) \cdot \frac{pp^{inv}(e) - \max_{e \in E}\left(pp^{\hat{inv}}(e)\right) + \log_{10}(pp_b)}{\log_{10}(pp_b)}$$

REWEIGHT


Word
RelatedTo
RelatedTo
God
Language
Rules &
Grammar Correction
1) a word is related to a god
2) a word is related to a language
LM
PP?
Sentence perplexity $PP(1) = 24.2$
Sentence perplexity $PP(2) = 4.0$
REWEIGHT
Weights

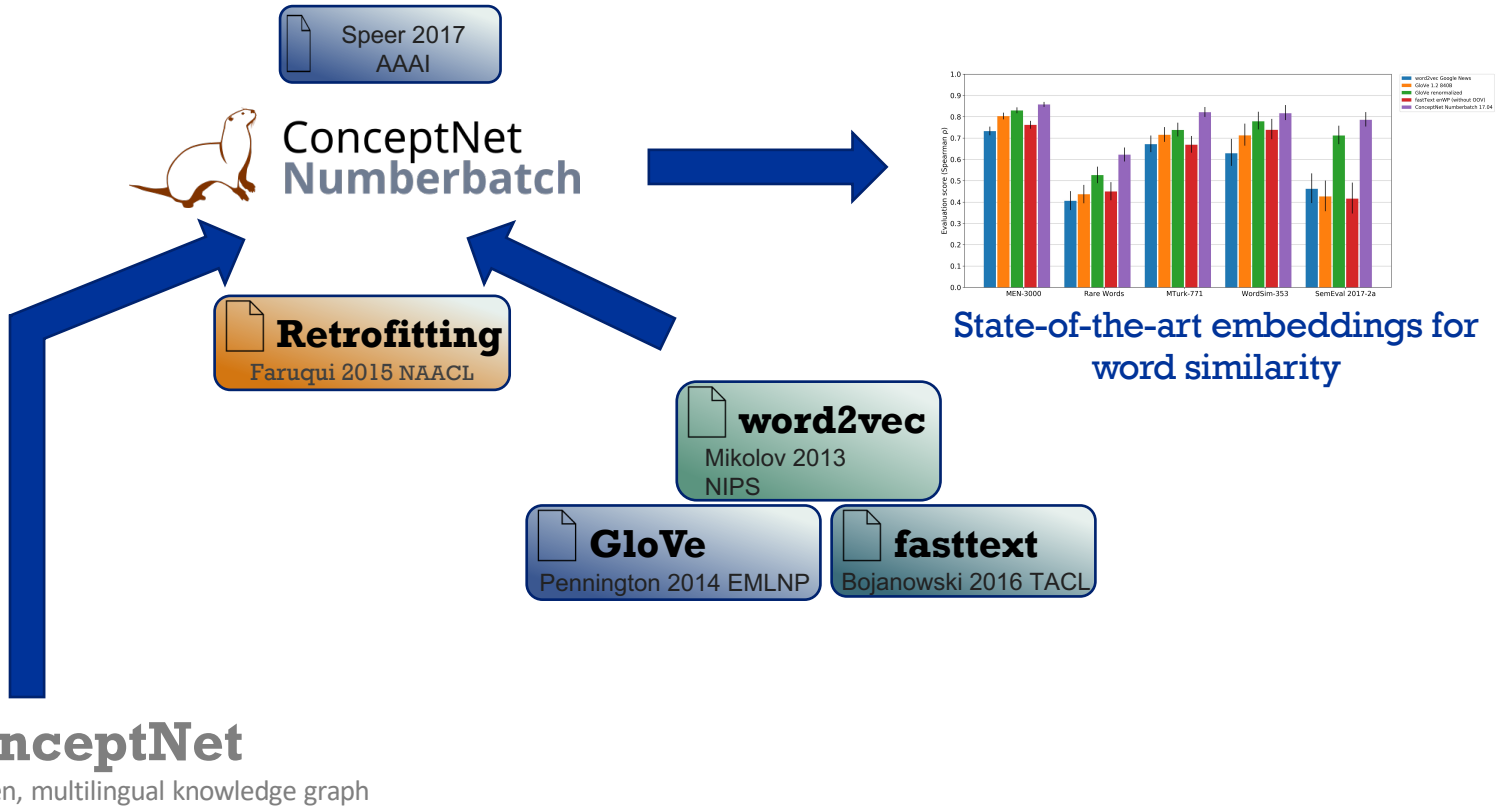
Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

data science

Evaluation

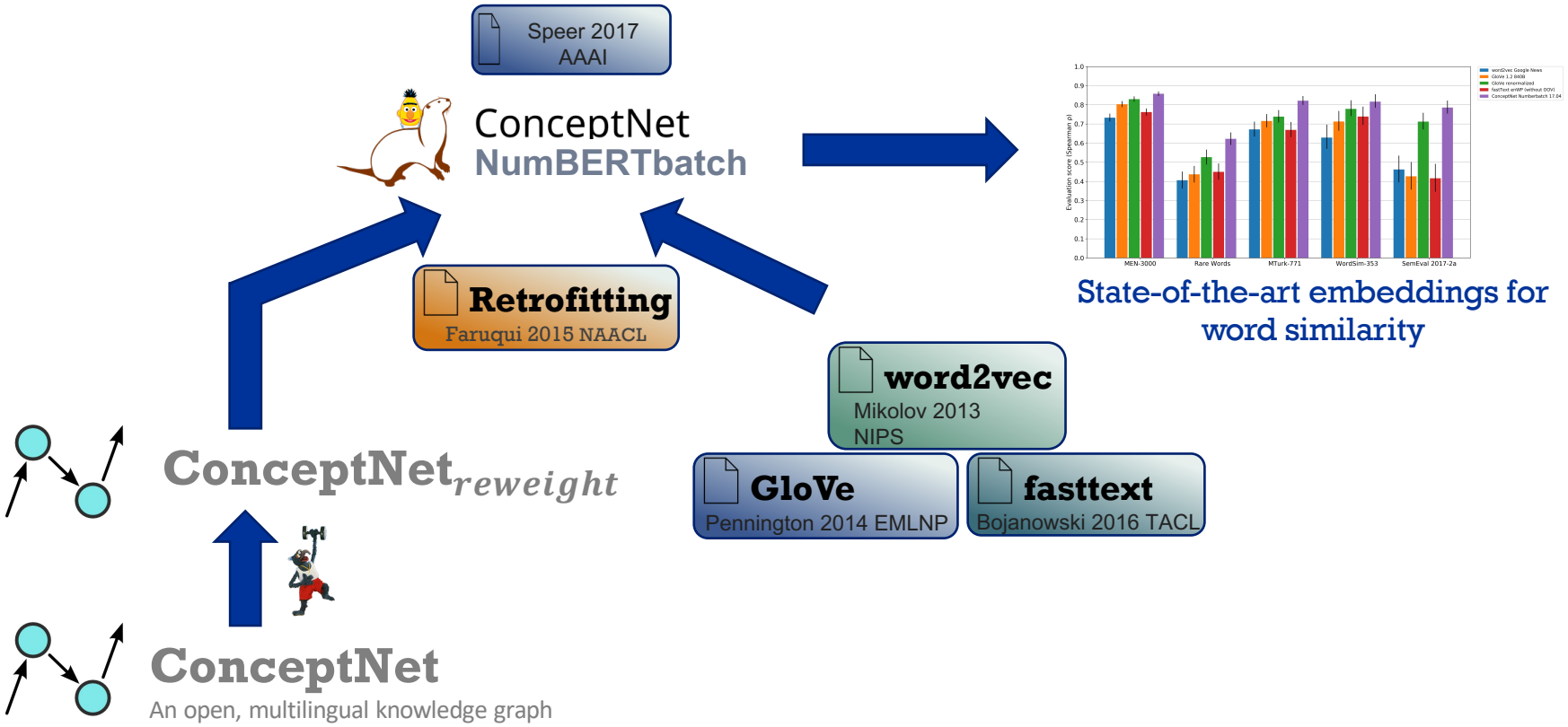
Improving Word Embeddings for Semantic Relatedness

Evaluation



State-of-the-art embeddings for word similarity

Evaluation



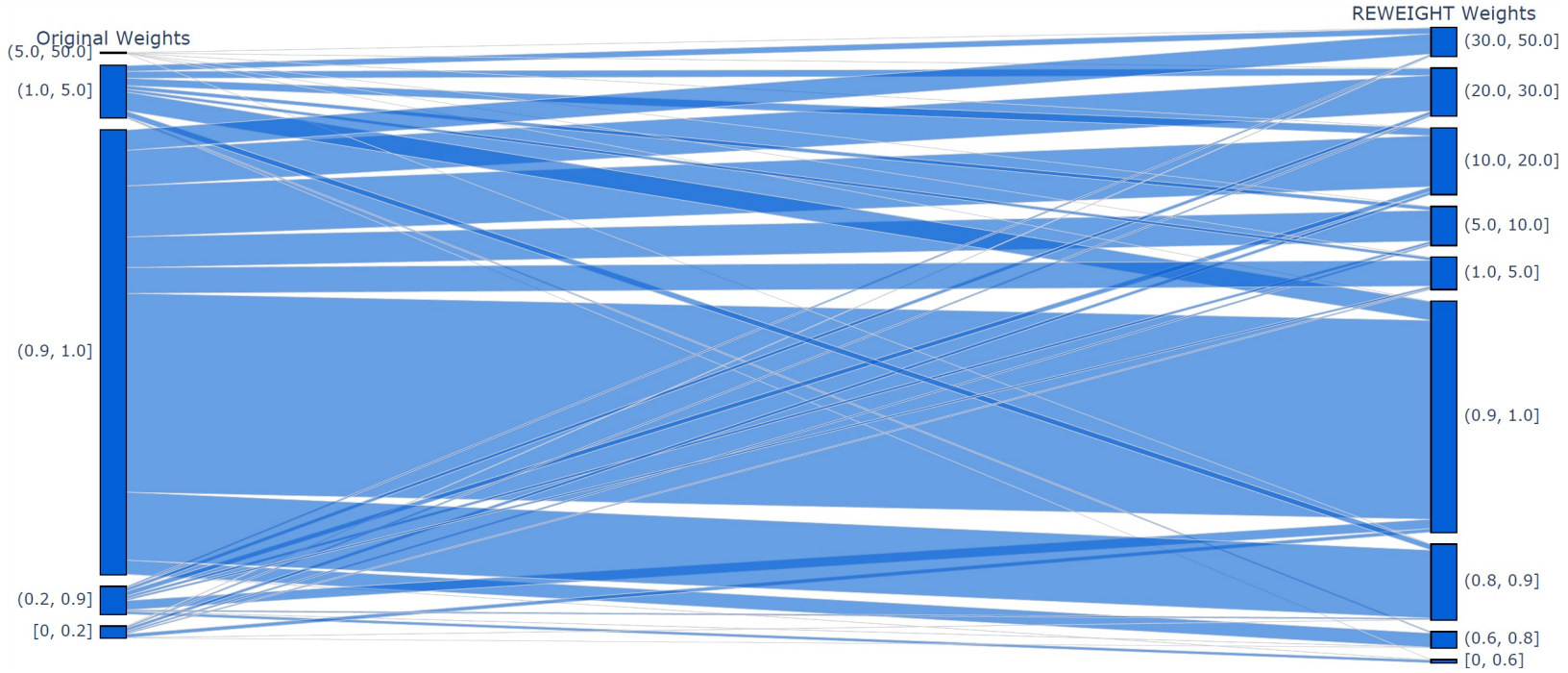
State-of-the-art embeddings for word similarity

Results

Embedding	MEN 3000	Rare Words (RW)	MTurk	WS353	SemEval	SimLex
Numberbatch	0.872	0.630	0.822	0.833	0.779	**0.633**
NumBERTbatch$_{light}$	0.877	**0.663***	0.827	0.840	**0.783**	**0.633**
NumBERTbatch$_{mod}$	**0.881***	0.651**	**0.828**	**0.845**	0.780	0.618

(Significance through Fischer's z-transformation with *p < 0.01, **p < 0.05)

ConceptNet Weight Transitions



ConceptNet Weight Transitions

- Success
  - Highly reasonable relations
    - E.g. „mathematics $RelatedTo$ geometry“ $(0.1 \rightarrow 34.8)$
  - Mildly questionable statements
    - E.g. „mathematical $SimilarTo$ unquestionable“ $(2.0 \rightarrow 0.88)$
- Problems
  - Specific keywords
    - E.g. „anthrax IsA disease“ $(2.8 \rightarrow 0.9)$ → Correct, but „anthrax“ not in BERT vocabulary

Further Experiments

- Conducted experiments
  - Reasonable perplexity border pp_b
  - Clipping outliers
  - Removing edges with low/high weights
  - Different LMs
  - Different KGs
  - Removing grammar correction
  - Random weight distribution

Future Work

- Assess different KGs in their own tasks
- Automatic tripel to sentence generation
- Assess out-of-vocabulary words (OOV)

Contact

- Further information in the paper:

  Omeliyanenko et al.
  "LM4KG: Improving Common Sense Knowledge Graphs with Language Models"
  ISWC 2020

- Code and Embeddings available under:
  https://github.com/JohannaOm/REWEIGHT

- Contact:

Janna Omeliyanenko
omeliyanenko@informatik.uni-wuerzburg.de
http://www.dmir.uni-wuerzburg.de/staff/omeliyanenko/