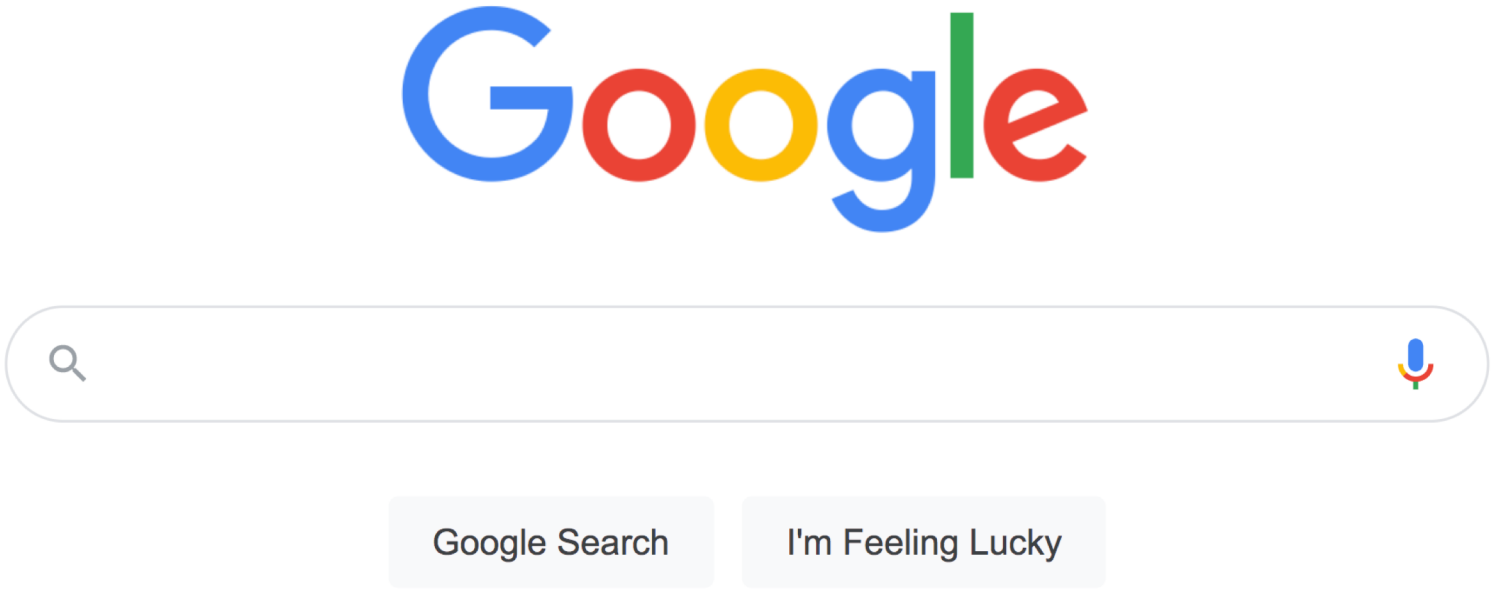
Evaluating Semantic Linking Capabilities of Engineering-Specific Word Embeddings Across Languages

Peter Karl Weinberger, 11th October 2021, Guided Research Final Presentation

Chair of Software Engineering for Business Information Systems (sebis)
Faculty of Informatics
Technische Universität München
wwwmatthes.in.tum.de

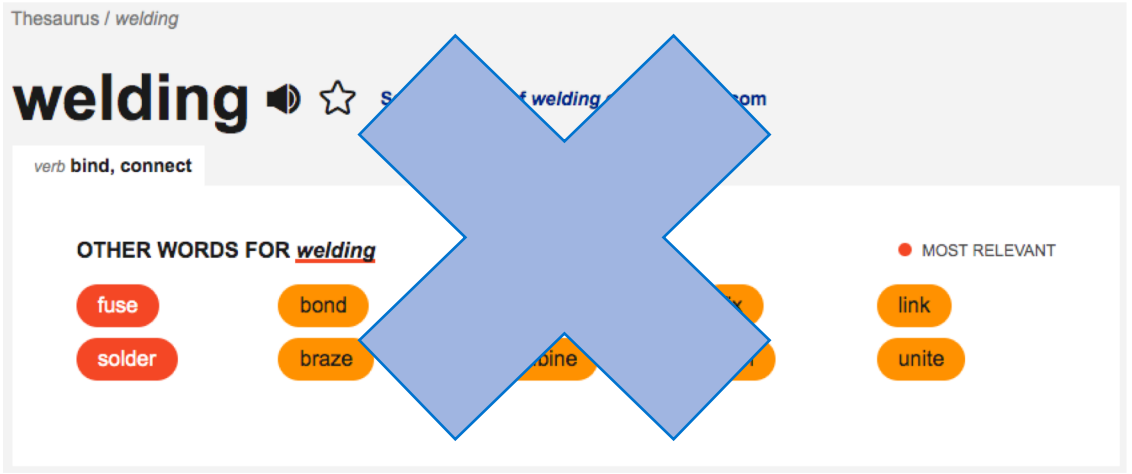
Outline

High-Level Idea



High-Level Idea

- Initial Situation
  - ~ 3 million engineering related articles in English or German
  - Word embedding as a tool to find topically similar words to certain keywords



High-Level Idea

- Initial Situation
  - ~ 3 million engineering related articles in English or German
  - Word embedding as a tool to find topically similar words to certain keywords

- Use Case
  - Identify – for the user – unknown technologies in multiple languages

  ⇒ Part of the research project Technology Scouting as a Service (TSaaS)

- Research Questions
  - How to merge engineering domain-specific word embeddings of different languages to ensure suitable semantic word comparisons between different languages?
  - How can this approach be adapted for semantic comparisons between engineering domain-specific articles of different languages?

Outline

High-Level Idea

Methods

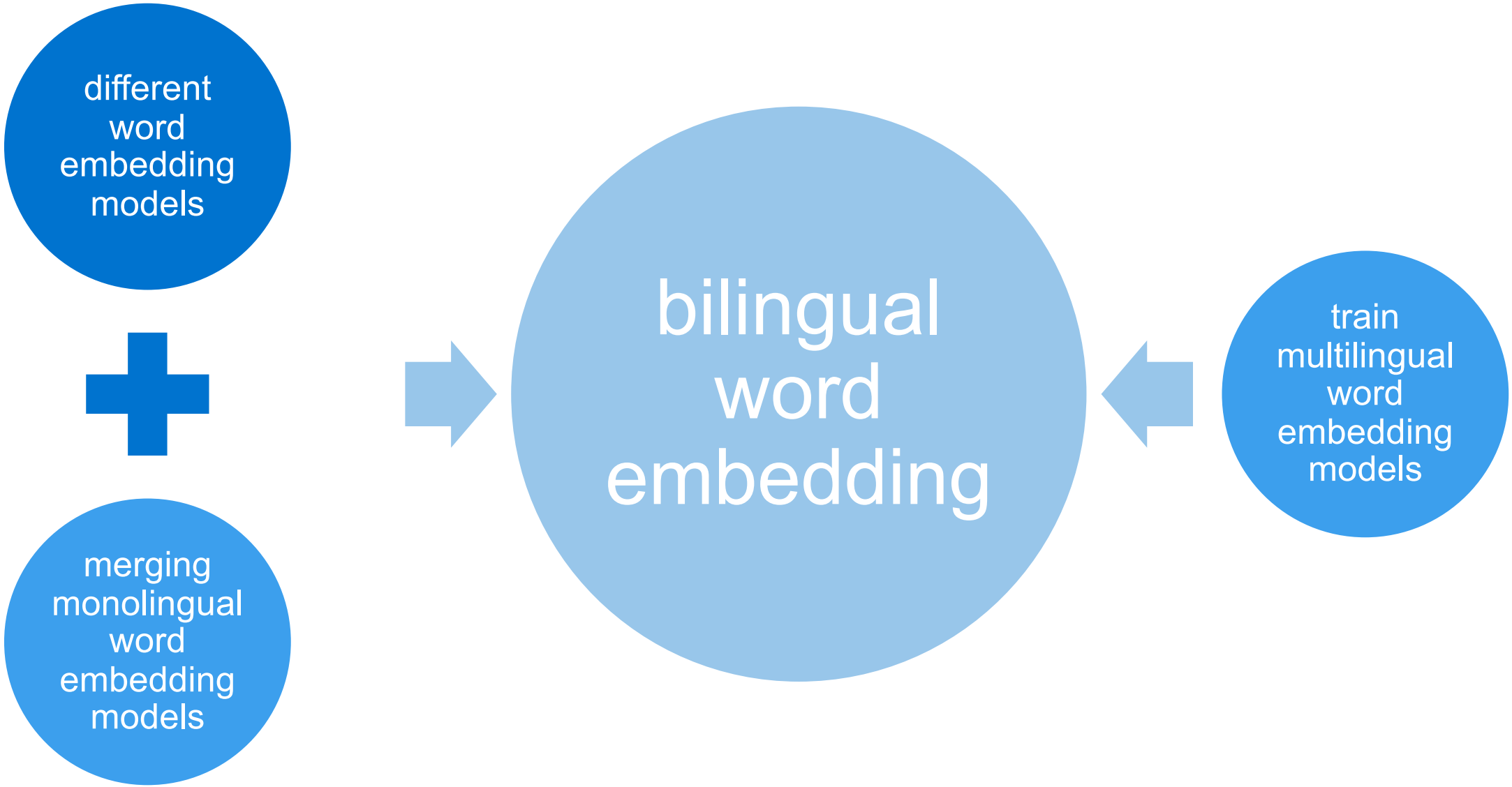
Experiments

Evaluation

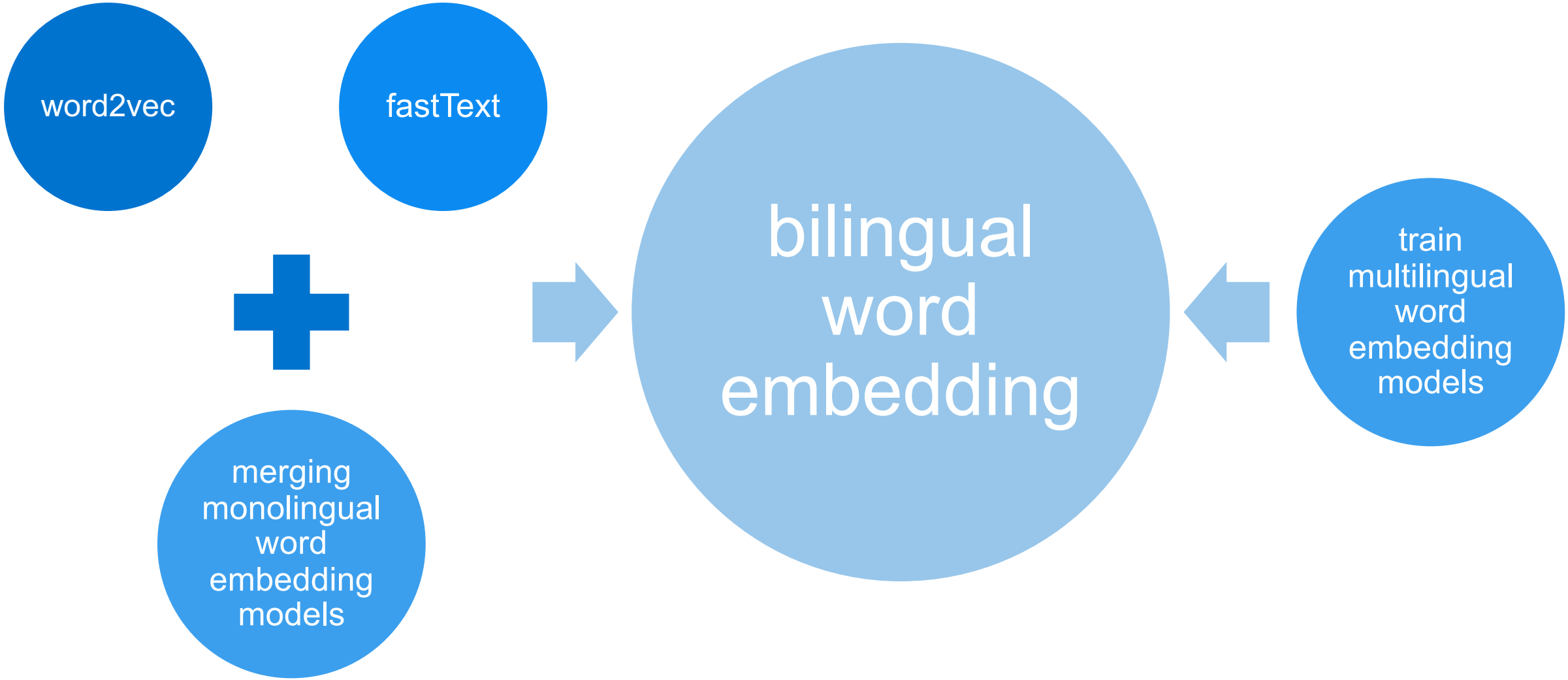
Findings

Future Work

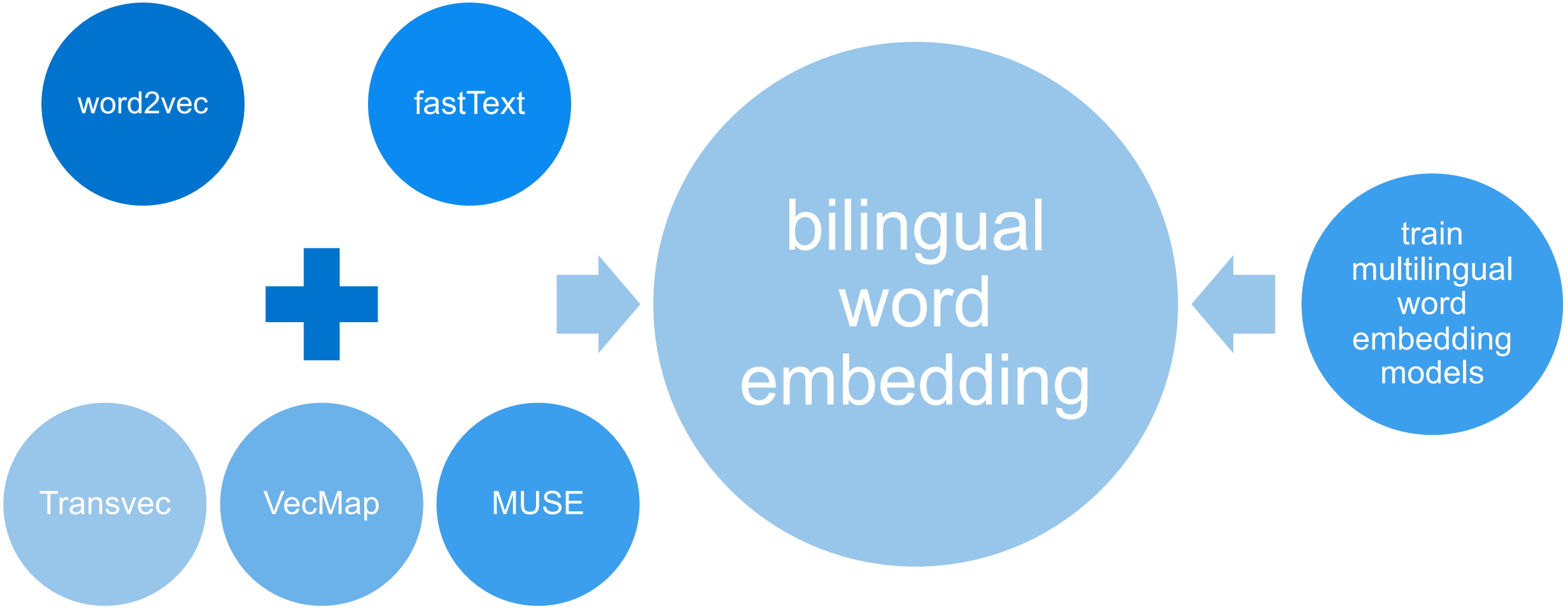
Methods



Methods



Methods



Gensim Library. “Word2vec embeddings“ https://radimrehurek.com/gensim/models/word2vec.html
Gensim Library. “FastText model“ https://radimrehurek.com/gensim/models/fasttext.html
big-o. “Transvec“ https://github.com/big-o/transvec
Facebook Inc.. “MUSE“ https://github.com/facebookresearch/MUSE
M. Artetxe. “VecMap“ https://github.com/artetxem/vecmap

Outline

High-Level Idea

Methods

Experiments

Evaluation

Findings

Future Work

Experiments

- Initial Experiments
  - Reduced data set (~1 mio. articles)

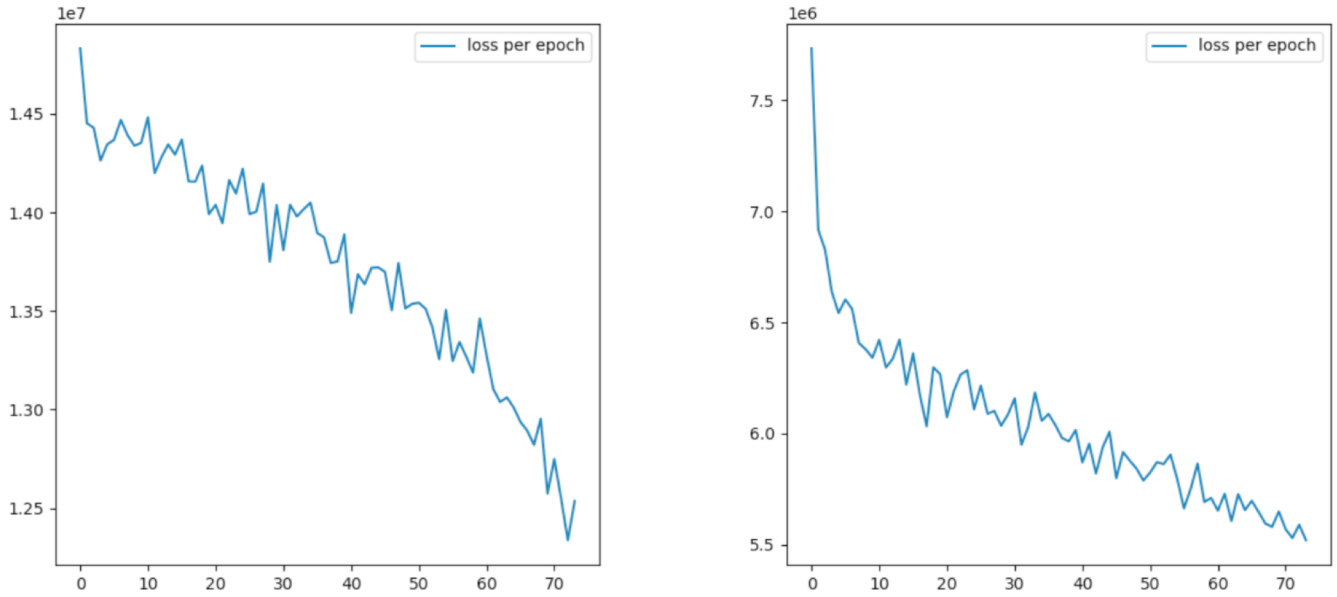


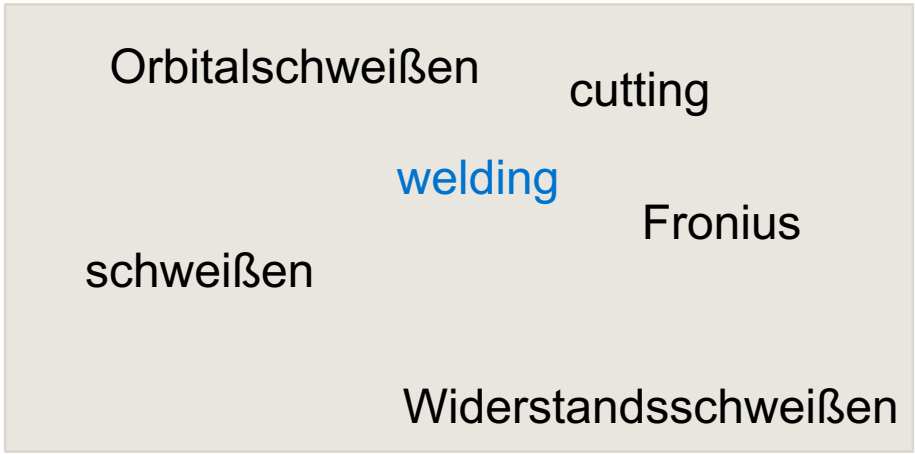
Figure 1: Loss of the word2vec model trained on the English corpus (left) and loss of the word2vec model trained on the German corpus (right). The *x*-axis indicates the number of epochs and the *y*-axis is the loss per epoch. Both models are trained on the reduced data set.

Experiments

- Initial Experiments
  - Reduced data set (~1 mio. articles)

Table 1: Ten nearest neighbors of the German word “schweißen” (English: “welding” in the two different models (left: fastText, right: word2vec; in descending order of the cosine similarity).

fastText	word2vec
nahtschweißen	laserschweißen
heftschweißen	laserschneiden
einschweißen	roboterschweißen
reibschweißen	punktschweißen
kaltschweißen	löten
tiefschweißen	verschweißen
trennschweißen	entgraten
tandemschweißen	stanzen
anschweißen	laserstrahlschweißen
handschweißen	biegen



Experiments

- Initial Experiments
  - Reduced data set (~1 mio. articles)
  - Map word2vec embeddings with Transvec and VecMap
  - Map fastText embeddings with MUSE
  - Utilization of handcrafted seed/learning dictionary (170 word pairs)

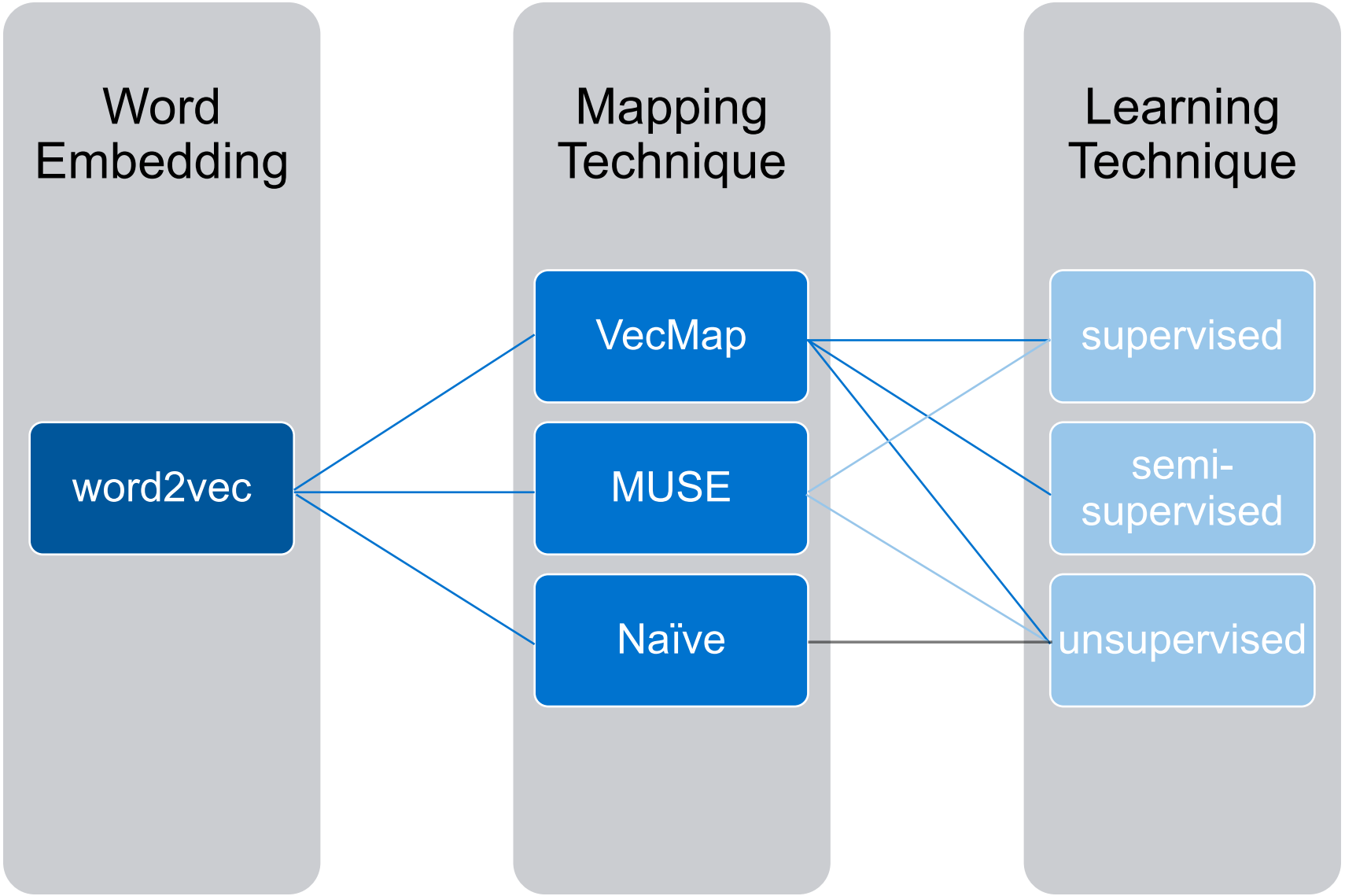
  ⇒ Transvec not applicable

  ⇒ fastText does not fit our use case

- Final Experiments
  - Full data set (~3 mio. articles)
  - Utilization scraped seed/learning dictionary (655 word pairs)
  - Applied findings from the initial experiments
    - Introducing suffixes to words
    - Increasing number of epochs while training word2vec models
    - Apply MUSE to word2vec models
  - Choosing models and mappings for evaluation
    - word2vec: vector size of 300, trained on 500 epochs

Übersetzungsbüro eLengua. *"Technisches Fachwörterbuch Deutsch – Englisch"* https://elengua.de/blog/technisches-woerterbuch-deutsch-englisch/

Final Experiments – Summary



Outline

High-Level Idea

Methods

Experiments

Evaluation

Findings

Future Work

Evaluation – Implementation

Comparative Intrinsic Evaluation

- Representative query words selected by domain experts
- Estimate k most similar words to each query word
- Domain experts must decide if translation is in set of neighborhood
- Neighborhood 1, 3, 5, and 10

$$score = \frac{\#\text{translation in k}}{\#\text{query words}}$$

Coherence

- Selection of query words
- Estimate k most similar words to each query word
- additionally a word (“intruder“) is added to the list which is not related to the query word
- Domain experts must decide which word does not fit the context
- Neighborhood 3, 5, and 10

$$score = \frac{\#\text{correctly class.}}{\#\text{query words}}$$

- In addition, we include a sanity check in the evaluation.

T. Schnabel, I. Labutov, D. Mimno, and T. Joachims. “Evaluation methods for unsupervised word embeddings.” https://www.aclweb.org/anthology/D15-1036/

Outline

High-Level Idea

Methods

Experiments

Evaluation

Findings

Future Work

Findings

- Semi-supervised VecMap and supervised MUSE (20 refinements) yields best results
- Sanity check delivers astonishing results: Models perform better on unseen data? Why?

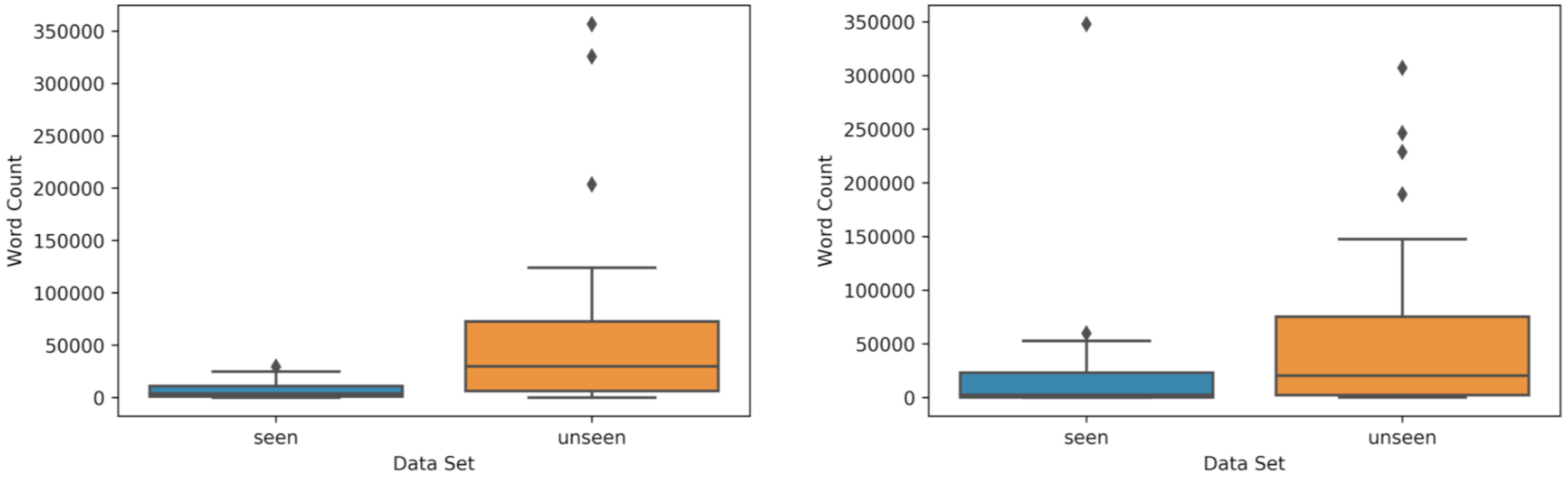


Figure 2: Comparison of the word count in the data sets used for both evaluation methods.

Findings

- Semi-supervised VecMap and supervised MUSE (20 refinements) yields best results
- Sanity check delivers astonishing results: Models perform better on unseen data? Why?

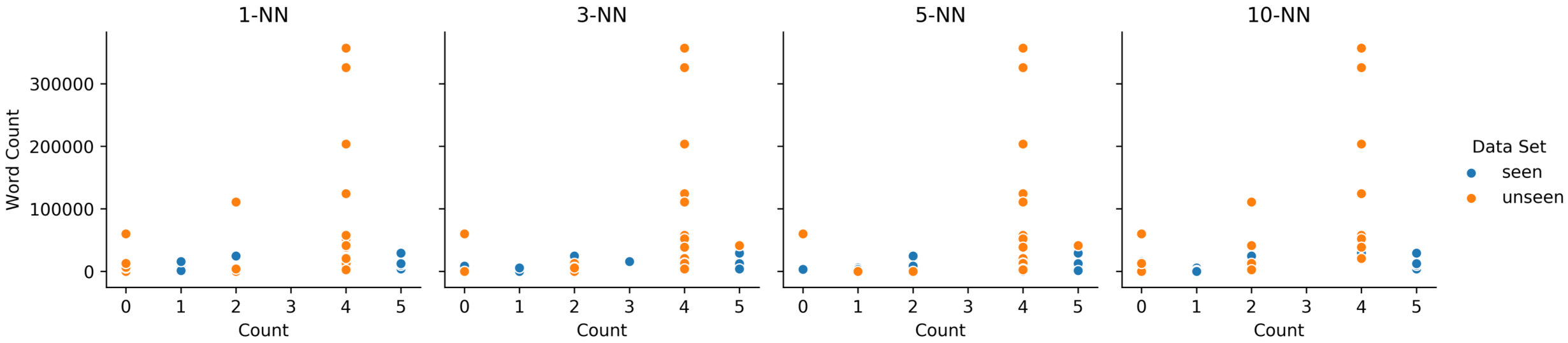


Figure 3: The count represents the number of times a direct translation of an evaluation word with an assigned word count is present in the respective neighborhood across all evaluated models. (Comparative Intrinsic Evaluation)

Findings

- Semi-supervised VecMap and supervised MUSE (20 refinements) yields best results
- Sanity check delivers astonishing results: Models perform better on unseen data? Why?

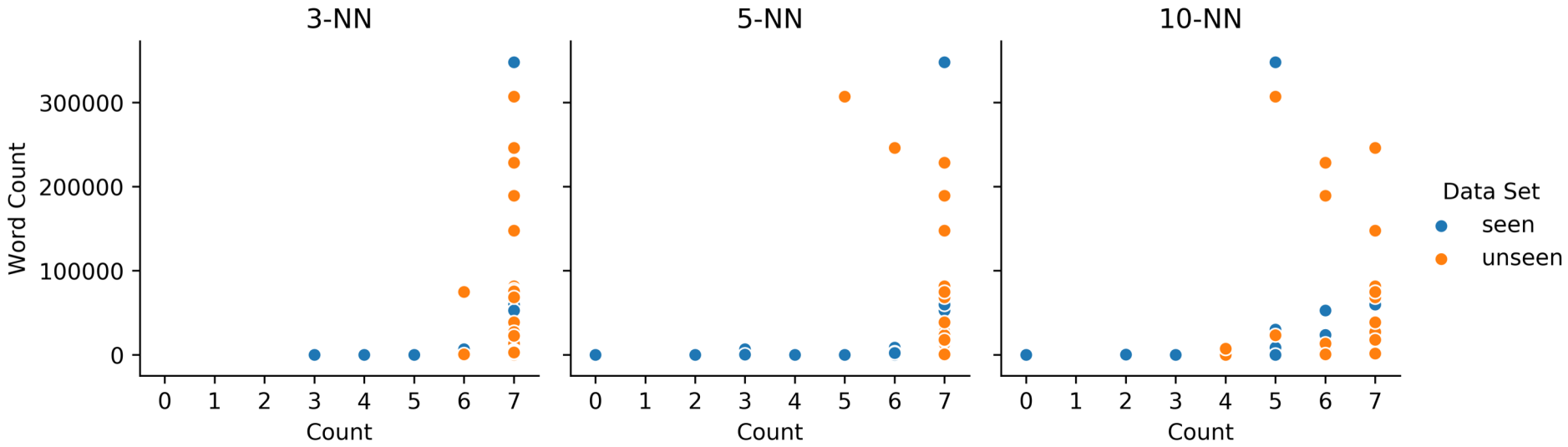
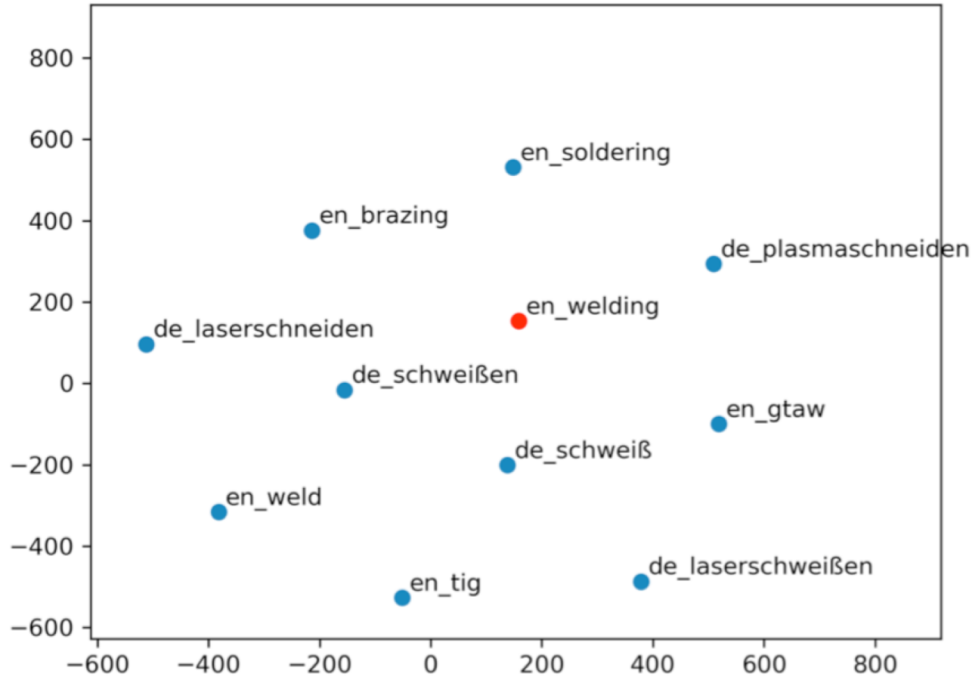


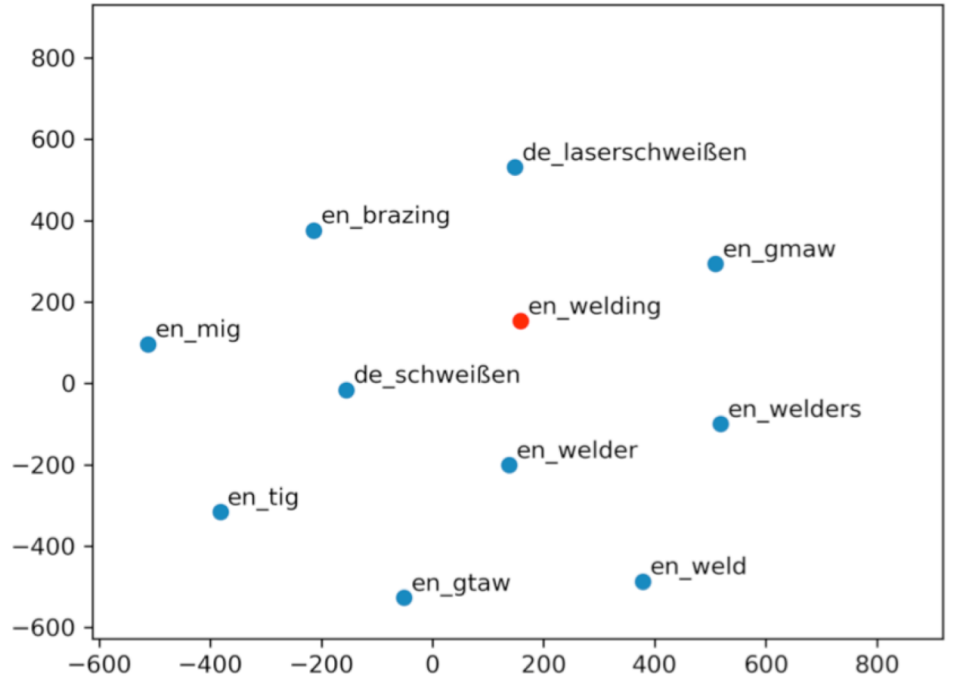
Figure 4: The count represents the number of times an intruder word is correctly classified in the respective neighborhood of an evaluation word with an assigned word count across all evaluated models. (Coherence)

Findings

- Semi-supervised VecMap and supervised MUSE (20 refinements) yields best results
- Word count in evaluation query words pervades the evaluation
- MUSE is heavily dependent on external evaluation data during training



(a) Semi-supervised VecMap.

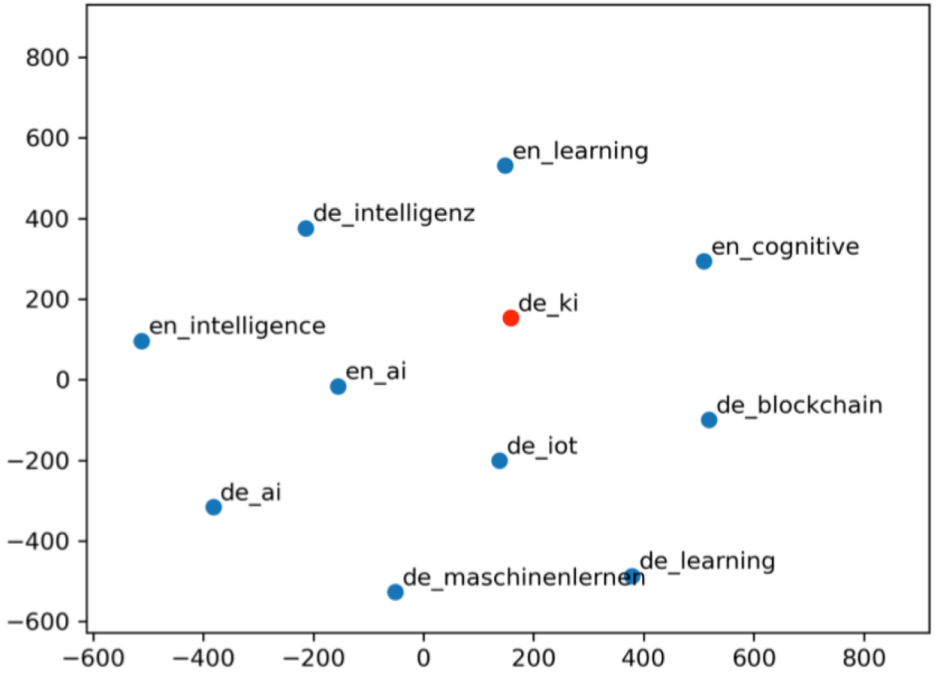


(b) Supervised MUSE (20 refinements).

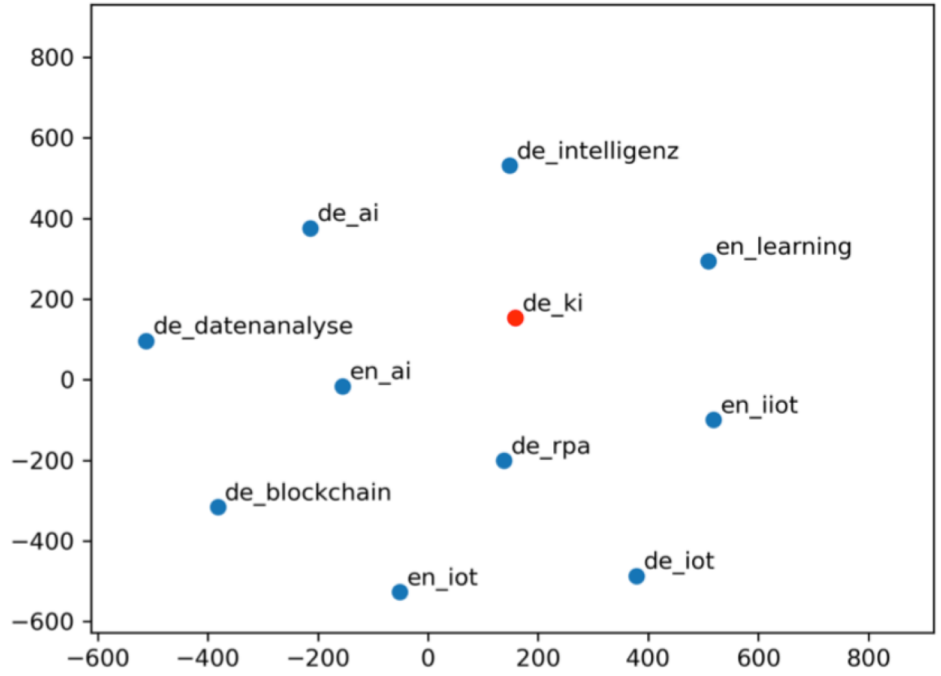
Figure 5: t-SNE projections of the embedded ten nearest neighbors (marked blue) of the word “en_welding” (marked red) into two dimensional space.

Findings

- Semi-supervised VecMap and supervised MUSE (20 refinements) yields best results
- Word count in evaluation query words pervades the evaluation
- MUSE is heavily dependent on external evaluation data during training



(a) Semi-supervised VecMap.



(b) Supervised MUSE (20 refinements).

Figure 6: t-SNE projections of the embedded ten nearest neighbors (marked blue) of the word “de_ki” (marked red) into two dimensional space.

Outline

High-Level Idea

Methods

Experiments

Evaluation

Findings

Future Work

Future Work

- Bilingual Document Embedding
  - Utilize supervised VecMap and embed documents via doc2vec
  - Utilize pre-trained multilingual sentence-BERT models and search for similar documents via Faiss
- Evaluation
  - Average evaluation results and exploit multiple domain experts e.g. target electrical engineering students for evaluation via an online survey
- Evaluation Data
  - Conduct evaluation on query words with below average word count, average word count, and above average word count as the word count of the query words drive the evaluation.

Gensim Library. “Doc2vec paragraph embeddings” https://radimrehurek.com/gensim/models/doc2vec.html
Facebook Inc. “Faiss“ https://github.com/facebookresearch/faiss
N. Reimers and I. Gurevych. “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks.” https://aclanthology.org/D19-1410.pdf
N. Reimers and I. Gurevych. “Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation.” https://aclanthology.org/2020.emnlp-main.365.pdf

Thank you for your attention!

It‘s time for questions.

Peter Karl Weinberger

Technische Universität München
Faculty of Informatics
Chair of Software Engineering for Business Information Systems

Boltzmannstraße 3
85748 Garching bei München