The space of meanings

Milica Gašić

Dialogue Systems and Machine Learning Group,
Heinrich Heine University Düsseldorf

In this lecture...

Understanding the user

Methods for obtaining word-vector embeddings

Core property of dialogue

- **Understanding** the user is a core property of dialogue
- Speech recogniser transforms speech into text but does not perform any understanding of the words
- Semantics concerns the meaning of words
- For now we will assume that the conversation is **goal-oriented** and that there exists an **ontology**

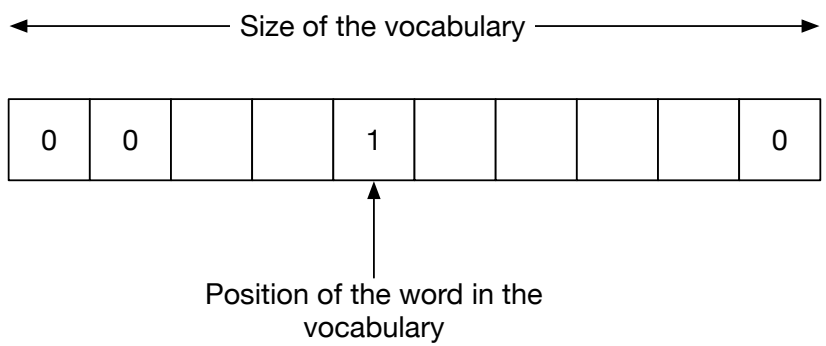
Understanding the user

The issue of understanding the user is two-fold:

- understanding the intention of the user (**dialogue act type**)
- understanding how what user is saying relates to the ontology (**slots** and **values**)

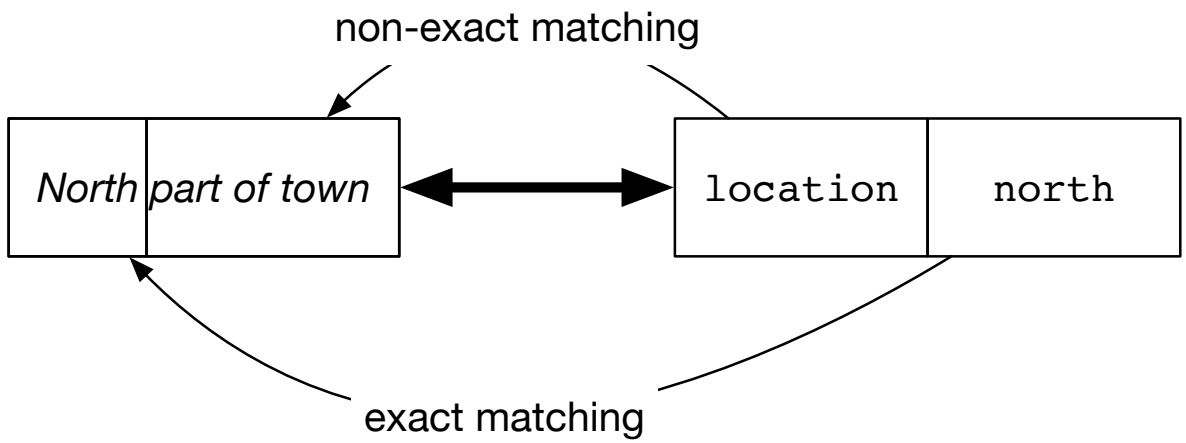
Symbolic representation

- We can represent dialogue concepts as symbols
- These can be human readable (eg. `id`, `name`, `location`, `title`)
- They can also be alphanumeric (eg. `xy345`) though this is not as common
- Internally symbolic representation equates to 1-hot representation
- This is also how we would represent all words in the vocabulary of the dialogue system



Symbolic representation

If we represent the concepts as symbols we need a process of matching words that the user is uttering to the symbols.



Natural Language Understanding is concerned with matching utterances to concepts.

Problems with symbolic representation

Non-exact matching

- We either need a lot of training data
- Alternatively, we need to exhaustively list all possible ways how something can be realised
  `location`: *part of town*, *area*, *located in*, ...
  This exhaustive list is called **semantic dictionary**.

Symbols are not related to each other

– They are equally distant from each other

- We can organise symbols in a graph, but this requires hand-coding.

Distributed representation

- We can represent words and concepts as vectors with a meaningful distance metric.
- Vectors that are close to each other have similar meaning.

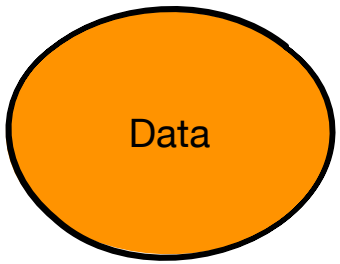
Distributional hypothesis

- *Die Bedeutung eines Wortes ist sein Gebrauch in der Sprache. (The meaning of words lies in their use.)*
  Wittgenstein [Wittgenstein, 1953]
- *All elements in a language can be grouped into classes whose relative occurrence can be stated exactly. However, for the occurrence of a particular member of one class relative to a particular member of another class it would be necessary to speak in terms of probability, based on the frequency of that occurrence in a sample* [Harris, 1954]
- **Distributional semantics:** data driven study of word meanings

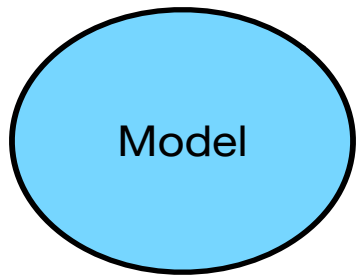
Distributional word vectors

- Real valued vectors that represent words
- Semantically related words have vectors close to each other
- Powerful tool for knowledge representation and learning

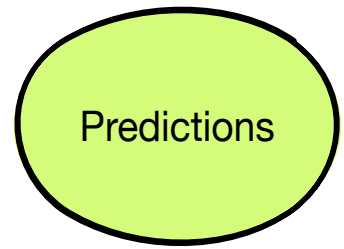
Machine learning for distributional semantics



- Text



- Unsupervised models



- Word vectors (word embeddings)

Evaluating word vectors

Intrinsic evaluation Cosine similarity

$$cos(w_i, w_j) = \frac{w_i^T w_j}{|w_i|^2 |w_j|^2}$$

Word analogies:

$$w(\texttt{queen}) \approx w(\texttt{king}) - w(\texttt{man}) + w(\texttt{woman})$$

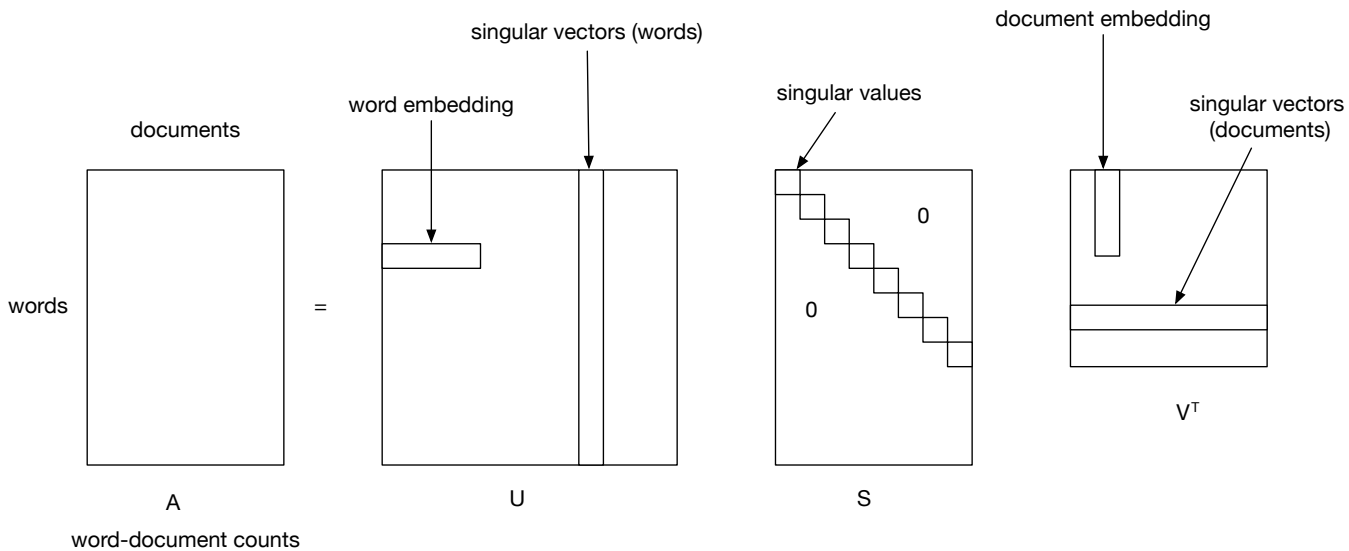
Extrinsic evaluation Performance on a downstream task

Use of word vectors in a downstream task

- Intrinsic evaluation is often not informative enough.
- When word vectors are updated in the downstream tasks this is the process of **fine-tuning**.
- If we have plentiful data for the downstream task then we can train word vectors from scratch.
- If we have limited data we need to perform fine-tuning but this in turn may lead to over-fitting.

Latent semantic analysis

- Compute word-document matrix A
- Use singular value decomposition $A = USV^T$
- It finds the most important directions of a data set, those directions along which the data varies the most.
- Rows of U (truncated to most important dimensions) are word vectors

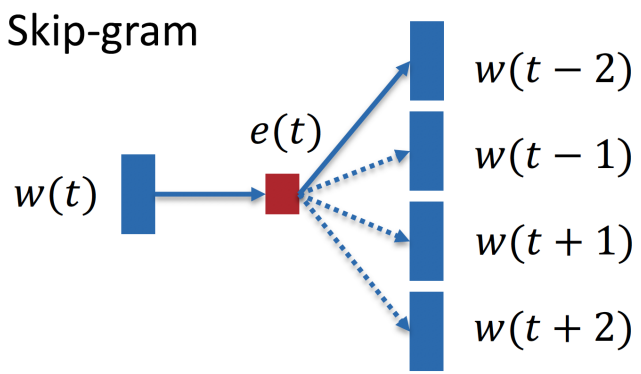


Pros vs cons

Model	Pros	Cons
LSA	Captures meaning	Strong assumptions

word2vec

- (Early) neural approach
- Learns given a word (target word) to predict its surrounding words (context words)
- Word embeddings are a by-product of this prediction task



$w(t-2)$	$w(t-1)$	$w(t)$	$w(t+1)$	$w(t+2)$	
a	refreshing	pool	of	cold	water

word2vec

- Treat the target word and a neighboring context word as positive examples.
- Randomly sample other words in the lexicon to get negative samples.
- Use a neural network classifier to distinguish those two cases.
- Use the weights of the neural network as the embeddings.

Why does word2vec work?

- Each word is initially represented as a 1-hot vector. The weight matrix W transforms this vector into a lower dimensional vector w.
- We have one such matrix W for the target words and one for context words or randomly selected words C, initially randomly initialised and then iteratively estimated.
- Similarity between two vectors is measured by a dot product $w^{\mathsf{T}}c$. A sigmoid function transforms this similarity into a probability.
- In the loss function we maximise similarity between the target word and the context and minimise the similarity between target word and the randomly selected words.

Analogical reasoning task

- Perform operations with vectors to answer questions $A - B + C \approx ?$
- What is to C in the same sense as B is to A?
- Closest vector according to cosine distance is taken as answer.

Relationship	Example	Example	Example
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
big - bigger	small: larger	cold: colder	quick: quicker
Copper - Cu	Zinc: Zn	Gold: Au	Uranium: Plutonium
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

Table 1: Examples for 300 dimensional embeddings, trained on 783M words with skip-gram model [Mikolov et al., 2013]

- This however may reveal biases and the ultimate source of such biases is the text itself [Eisenstein, 2019].

Pros vs cons

Model	Pros	Cons
LSA	Captures meaning	Strong assumptions
word2vec	Intuitive	Local context

GloVe - Global Vectors

- GloVe considers global context
- Utilizes a co-occurrence matrix to capture global statistics
- Word co-occurrence probability ratios have potential to encode meaning.

Probability and ratio	k = solid	k = gas	k = water	k = fashion
$P(k\|\text{ice})$	1.9×10^{-4}	6.6×10^{-4}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|\text{steam})$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|\text{ice}) / P(k\|\text{steam})$	8.9	8.5×10^{-2}	1.36	0.96

Table 2: In the ratio noise from non-discriminative words (water, fashion) cancels out. Large values (much greater than 1) correlate well with ice, and small values (much less than 1) correlate well with steam [Pennington et al., 2014].

GloVe

- Word embeddings w_i are found by minimising the sum of squares:

$$J = \Sigma_{i,j} \frac{X_{ij}}{X_{\max}} (w_i \tilde{w}_j^{\mathsf{T}} - log X_{ij})^2$$

  where X_{ij} is the co-occurrence count of word represented by vector w_i occurring in the context of word represented by vector $\tilde{w}_j$ and $X_{\max}$ the largest co-occurrence
- Goal: Learn word vectors such that their product equals the log of their co-occurrence probability

Pros vs cons

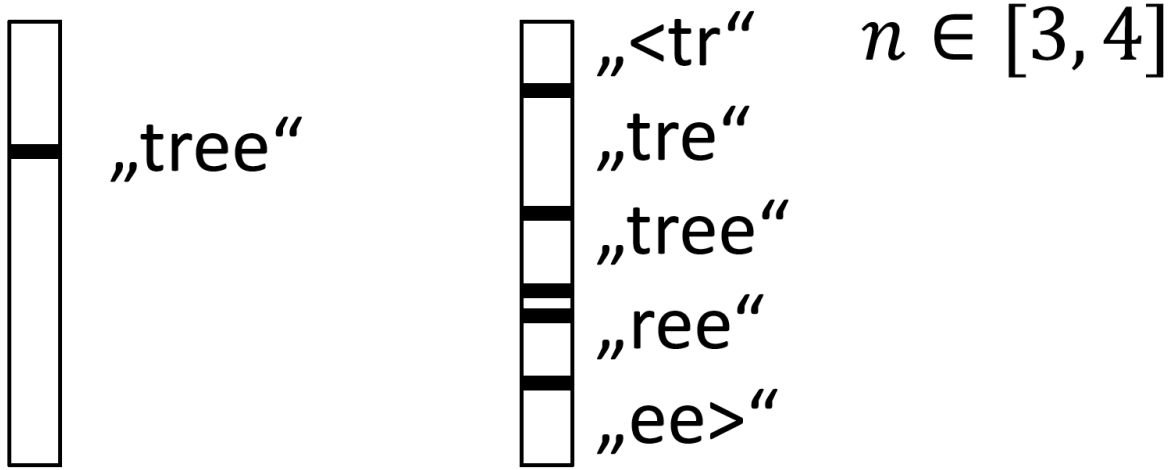
Model	Pros	Cons
LSA	Captures meaning	Strong assumptions
word2vec	Intuitive	Local context
Glove	Global context	Doesn't handle OOV

fastText

- Word2vec and GloVe consider words as smallest unit
- fastText sees words as being composed of character n-grams
- Generates better embeddings for rare words
- Can construct vectors for unseen (OOV) words

fastText

- Each word in fasttext is represented as itself plus a bag of constituent n-grams, with special boundary symbols $<$ and $>$ added to each word.
- Then a word2vec embedding is learned for each constituent n-gram, and the word embedding is represented by the sum of all of the embeddings of its constituent n-grams.



fastText (fasttext.cc)

Importance of subword information.

typo

word2vec:

```
Query word? accomodation
    sunnhordland 0.775057
    accomodations 0.769206
    administrational 0.753011
    laponian 0.752274
    ammenities 0.750805
    dachas 0.75026
    vuosaari 0.74172
    hostelling 0.739995
    greenbelts 0.733975
    asserbo 0.732465
```

fastText:

```
Query word? accomodation
    accomodations 0.96342
    accommodation 0.942124
    accommodations 0.915427
    accommodative 0.847751
    accommodating 0.794353
    accomodated 0.740381
    amenities 0.729746
    catering 0.725975
    accomodate 0.703177
    hospitality 0.701426
```

Pros vs cons

Model	Pros	Cons
LSA	Captures meaning	Strong assumptions
word2vec	Intuitive	Local context
Glove	Global context	Doesn't handle OOV
fastText	Handles OOV	Not contextual

Language models

- A language model assigns a probability to sequences of words

$$p(t_1, \ldots, t_N) = p(t_1)p(t_2|t_1)p(t_3|t_1, t_2) \cdots p(t_N|t_{N-1} \ldots, t_1), \tag{1}$$

  where N is the length of the sequence.
- The simplest language model is an **n-gram** language model, which takes fixed context of n preceding tokens into account, and can simply be estimated based on the frequencies of occurrences in the corpus:

$$p(t_k|t_{k-1}, \ldots, t_1) \approx p(t_k|t_{k-1}, \ldots, t_{k-(n-1)}), \tag{2}$$

  where t_i is a word on the ith position.

Neural language models

- Recurrent neural networks allow consideration of longer contexts
- Hidden layers of these models can serve as a context-dependent embedding
- This means that we obtain the embedding only when we present the context

ELMo - Embeddings from language models

- Deep contextualized word representation
- Learned function of internal states of a deep bidirectional language model
- Each token representation is a function of the entire input sentence

ELMo

- Tokens are represented as a linear combination of hidden layers
- The weights of the linear combination are task specific, while the biLM parameters are fixed
- Higher layers seem to capture semantics, lower layers syntactics
- Typically used as additional features

Dealing with polysemous words

Embedding	Source	Nearest Neighbour
Glove	**play**	playing, game, games, played players, plays, player, Play, football, multiplayer
ELMo	Chico Ruiz made a spectacular **play** on Alusik s grounder . . .	Kieffer , the only junior in the group , was commended for his ability to hit in the clutch , as well as his all-round excellent **play**
	Olivia De Havilland signed to do a Broadway **play** for Garson . . .	. . . they were actors who had been handed fat roles in a successful **play** , and had talent enough to fill the roles competently , with nice understatement .

Table 3: Nearest neighbors of token play using GloVe or ELMo [Peters et al., 2018]

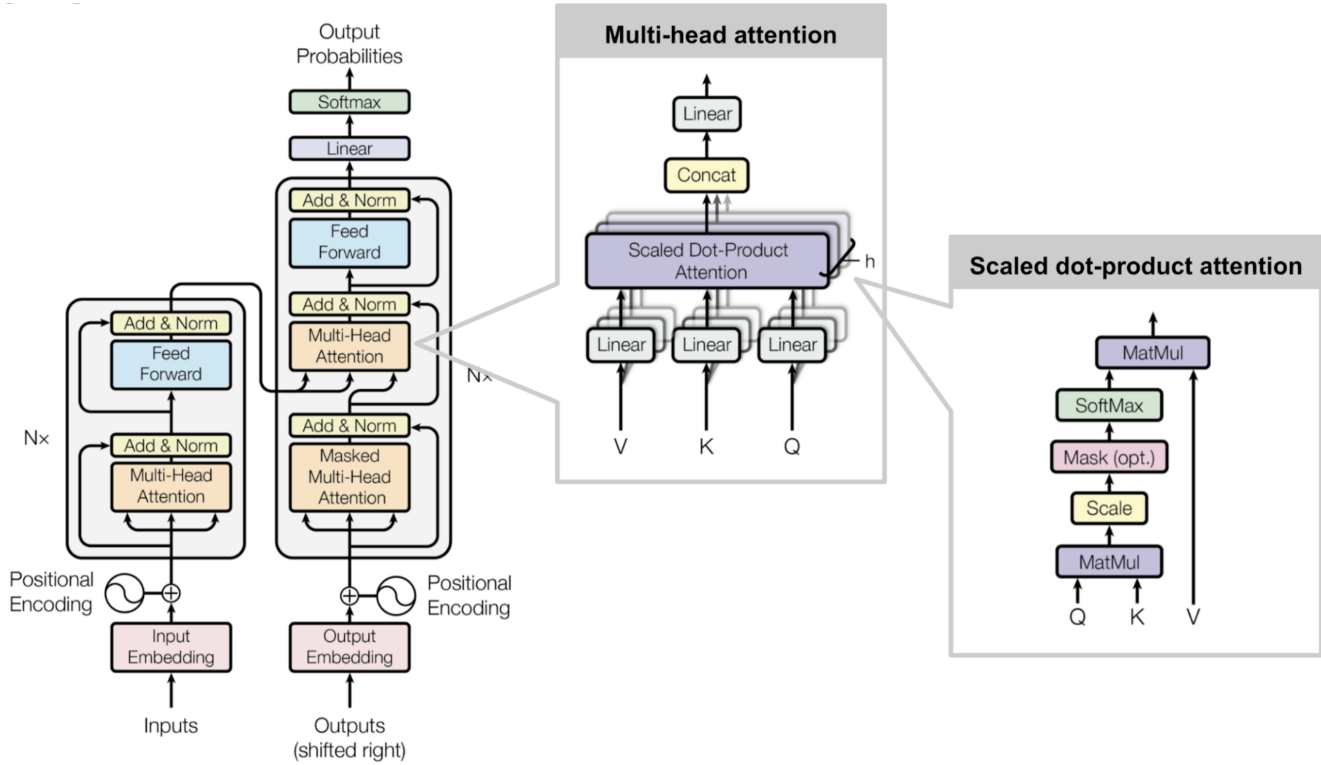
ELMo

- Conditional language models can only be trained left-to-right or right-to-left, since bidirectional conditioning would allow each word to indirectly see itself, and the model could trivially predict the target word.
- For that reason they do not adequately consider context from both directions.

Pros vs cons

Model	Pros	Cons
LSA	Captures meaning	Strong assumptions
word2vec	Intuitive	Local context
Glove	Global context	Doesn't handle OOV
fastText	Handles OOV	Not contextual
ELMo	Contextual	Not truly bidirectional

Transformer [Vaswani et al., 2017]



BERT - Bidirectional Encoder Representation from Transformers

- The backbone of BERT are stacked transformers
- They are trained on a **masked** language modelling task
- This means that with some probability words in the input are replaced with a generic token `[MASK]`
- Transformer allows for output to be conditioned on context coming from both directions
- In addition to this, the model is trained on the next sentence prediction to obtain sentence embeddings

BERT

- BERT-base is trained on more than 3,000M words
- Still, BERT embeddings on their own are not used
- They are normally *fine-tuned* for specific task

Fine-tuning BERT embeddings

- Masked language modelling task of the last layer (or the next-sentence prediction) can be replaced with another task.
- For example, instead of next sentence prediction we can have prediction of an answer to the question.
- In this way BERT embeddings become geared towards a QA task
- In the same way, we can fine-tune BERT for a particular dialogue sub-task

Pros vs cons

Model	Pros	Cons
LSA	Captures meaning	Strong assumptions
word2vec	Intuitive	Local context
Glove	Global context	Doesn't handle OOV
fastText	Handles OOV	Not contextual
ELMo	Contextual	Not truly bidirectional
BERT	Bidirectional	Computationally expensive

Summary

- Understanding as core property of dialogue
- Symbolic vs distributed representation
- Distributional hypothesis
- Static embeddings
- Contextualised embeddings

Next lecture

- Spoken language understanding

References I


Eisenstein, J. (2019).
Introduction to Natural Language Processing.
Adaptive Computation and Machine Learning series. MIT Press.


Harris, Z. S. (1954).
Distributional structure.
WORD, 10(2-3):146–162.


Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013).
Efficient estimation of word representations in vector space.
In Bengio, Y. and LeCun, Y., editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

References II


Pennington, J., Socher, R., and Manning, C. D. (2014).
Glove: Global vectors for word representation.
In *In EMNLP*.


Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018).
Deep contextualized word representations.
In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

References III


Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017).
Attention is all you need.
In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.


Wittgenstein, L. (1953).
Philosophische Untersuchungen.

Credits

We thank Michael Heck for sharing his slides The world of word embeddings