Towards German Word Embeddings: A Use Case with Predictive Sentiment Analysis

Eduardo Brito
eduardo.alfredo.brito.chacon@iais.fraunhofer.de

Fraunhofer Institute for
Intelligent Analysis and Information Systems IAIS

12th June 2017

Word Embeddings

- Words are highly related to the contexts where they appear

Word Embeddings

- Words are highly related to the contexts where they appear

... German musician and composer of the Baroque ...

Word Embeddings

- Words are highly related to the contexts where they appear

... German musician and composer of the Baroque ...

Bach? Any other German Baroque musician and composer?

Word Embeddings

- Words are highly related to the contexts where they appear

... German musician and composer of the Baroque ...

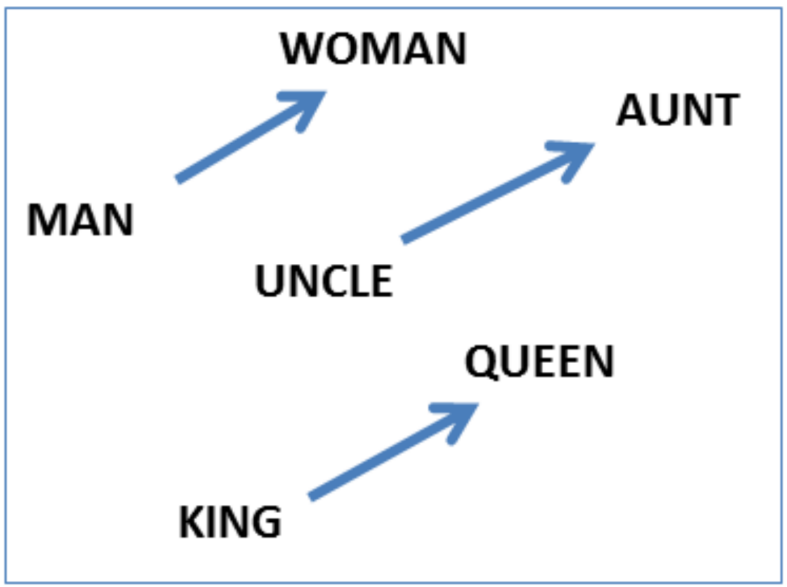
Bach? Any other German Baroque musician and composer?

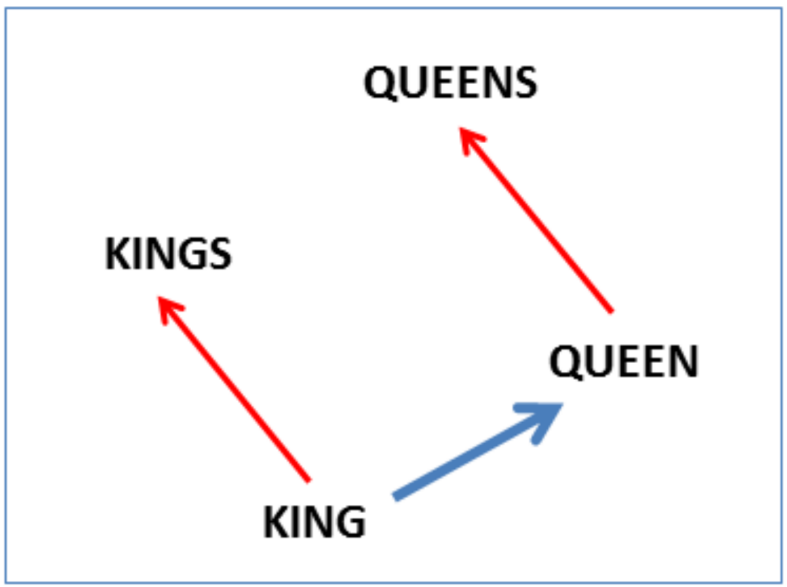
Idea

- Represent words as vectors
- Similar words are close to each other in the vector space
- Train word representations with an artificial neural network
- Use word vectors for any NLP task requiring language modeling

Word Embeddings: Vector Operations

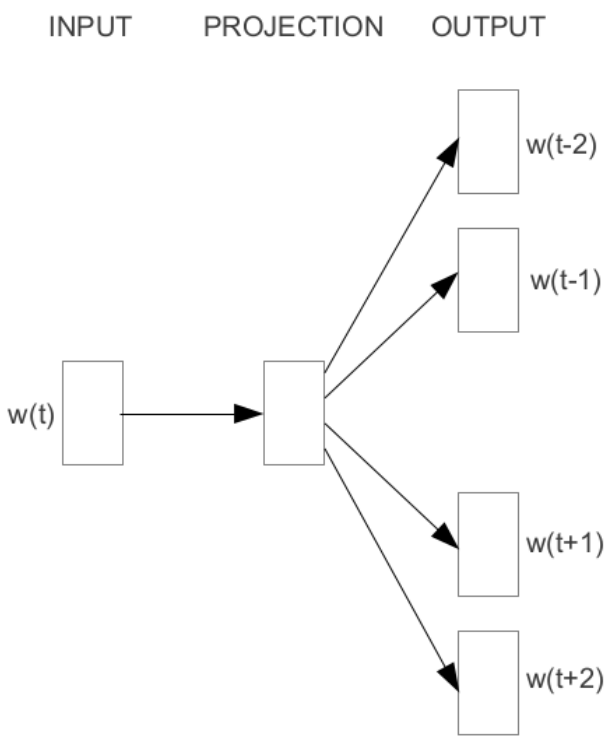
- Word embeddings capture both semantic and syntactic information
- Semantic problems can be reduced to a geometry problems





Source: Linguistic Regularities in Continuous Space Word Representations (Mikolov et al. 2013)

Continuous Skip-gram Model (SG)

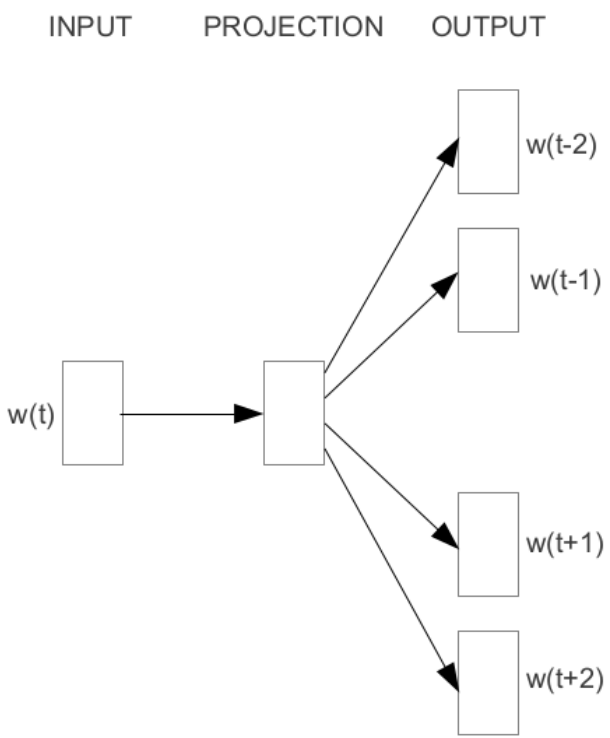


Source: Efficient Estimation of Word Representations in Vector Space (Mikolov et al. 2013)

Input layer

Input word as one-hot vector

Continuous Skip-gram Model (SG)



Source: Efficient Estimation of Word Representations in Vector Space (Mikolov et al. 2013)

Input layer

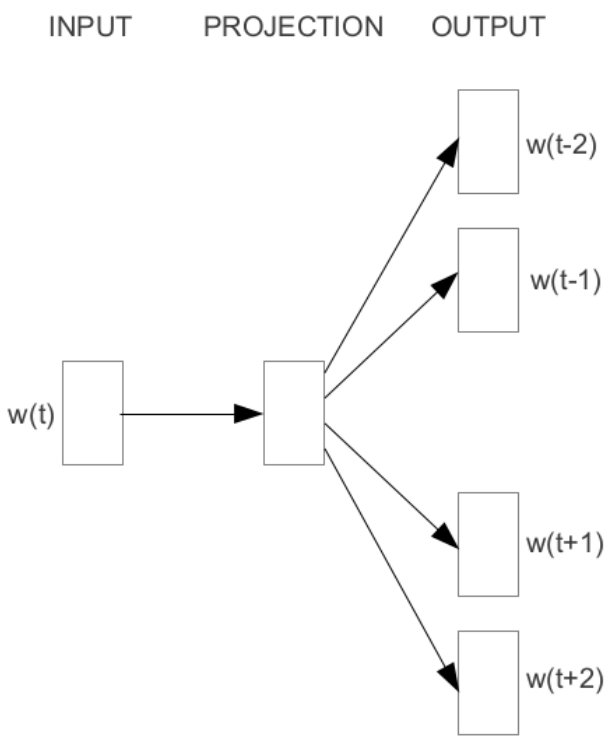
Input word as one-hot vector

Hidden layer

Projection to word embedding:

- One neuron per word in vocabulary
- Neuron weights are our word embeddings!

Continuous Skip-gram Model (SG)



Source: Efficient Estimation of Word Representations in Vector Space (Mikolov et al. 2013)

Input layer

Input word as one-hot vector

Hidden layer

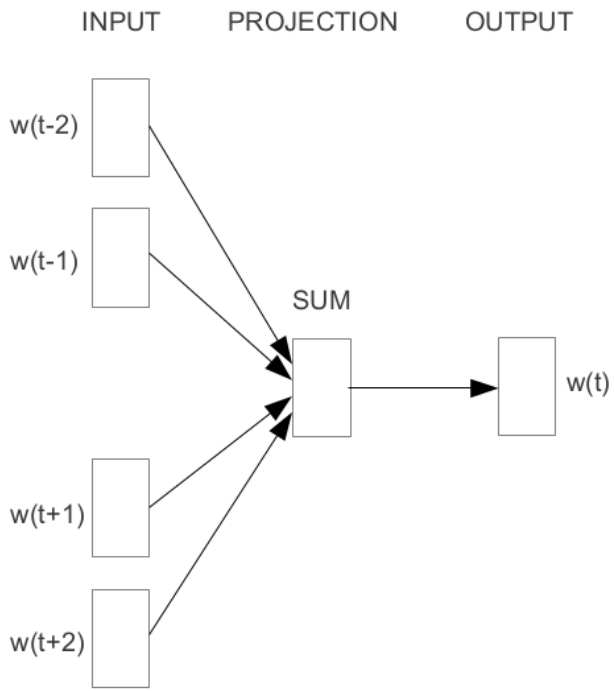
Projection to word embedding:

- One neuron per word in vocabulary
- Neuron weights are our word embeddings!

Output layer

- Context embeddings
- Dot product $\approx$ context probability
- Simplification: negative sampling or hierarchical softmax

Continuous Bag-of-Words Model (CBOW)

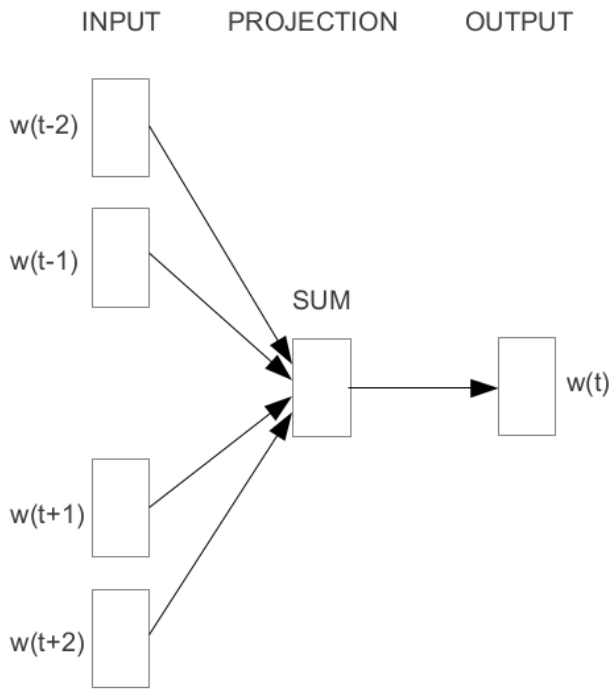


Source: Efficient Estimation of Word Representations in Vector Space (Mikolov et al. 2013)

Input layer

Context words as one-hot vectors

Continuous Bag-of-Words Model (CBOW)



Source: Efficient Estimation of Word Representations in Vector Space (Mikolov et al. 2013)

Input layer

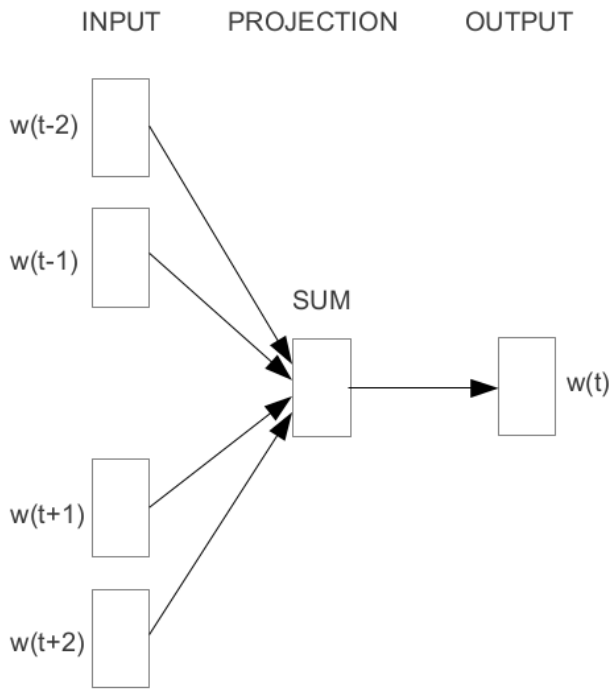
Context words as one-hot vectors

Hidden layer

Projection to word embeddings (from context):

- Word vectors are summed for predicting focused word

Continuous Bag-of-Words Model (CBOW)



Source: Efficient Estimation of Word Representations in Vector Space (Mikolov et al. 2013)

Input layer

Context words as one-hot vectors

Hidden layer

Projection to word embeddings (from context):

- Word vectors are summed for predicting focused word

Output layer

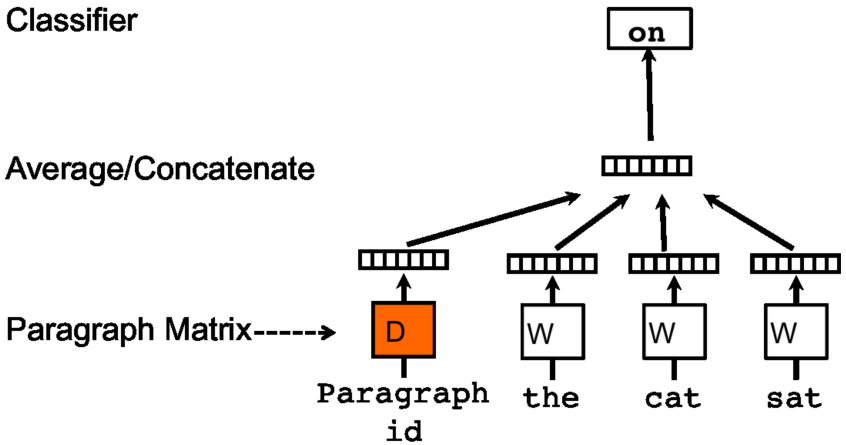
- Dot product $\approx$ word probability
- Simplification: negative sampling or hierarchical softmax

Paragraph Vector Model

Extension of CBOW and SG for document embeddings

Paragraph Vector Model

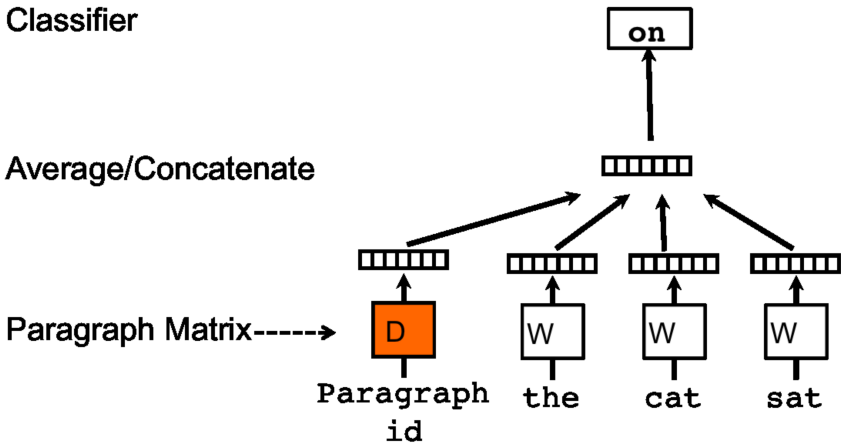
Extension of CBOW and SG for document embeddings



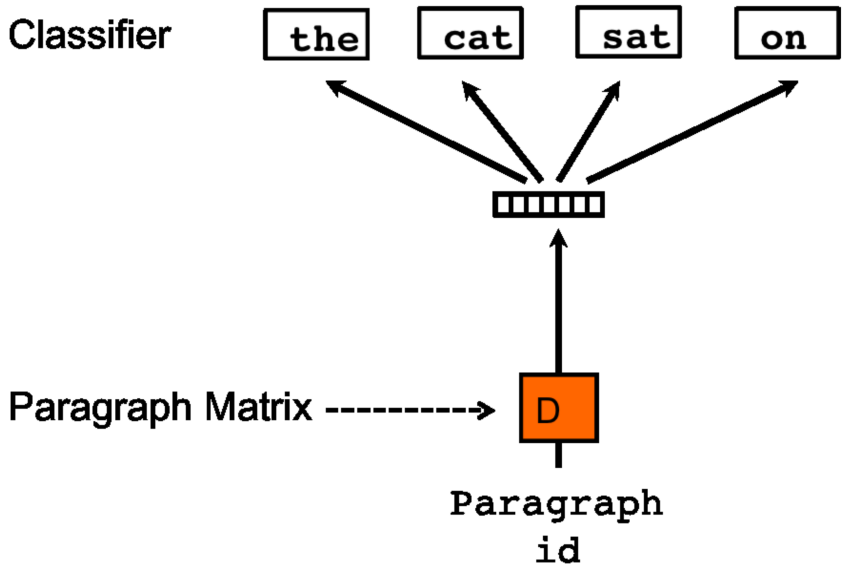
Distributed Memory version of Paragraph Vector (PV-DM)

Paragraph Vector Model

Extension of CBOW and SG for document embeddings



Distributed Memory version of Paragraph Vector (PV-DM)



Distributed Bag-of-words version of Paragraph Vector (PV-DBOW)

Source: Distributed Representations of Sentences and Documents (Le and Mikolov 2014)

Training German Word Embeddings

Motivation

- Most experiments on word embeddings only in English
- Hyper-parameter recommendations: similar for German?

Training German Word Embeddings

Motivation

- Most experiments on word embeddings only in English
- Hyper-parameter recommendations: similar for German?

Training German Word Embeddings

Motivation

- Most experiments on word embeddings only in English
- Hyper-parameter recommendations: similar for German?

Training corpus: SdeWaC

- Web-crawled text collection from .de domain
- 44,084,442 sentences
- 846,159,403 word tokens
- 1,094,902 different word types
- Simple preprocessing: tokenization, lowercasing, shuffling

Training German Word Embeddings

Motivation

- Most experiments on word embeddings only in English
- Hyper-parameter recommendations: similar for German?

Training corpus: SdeWaC

- Web-crawled text collection from .de domain
- 44,084,442 sentences
- 846,159,403 word tokens
- 1,094,902 different word types
- Simple preprocessing: tokenization, lowercasing, shuffling

Approach

- Original Google `word2vec` software package
- Train 11 different models (different hyper-parameters)
- Word vector size: 300

Training German Word Embeddings: Intrinsic Evaluation

Evaluation: semantic relatedness

- Gur350 dataset
  - 350 pairs of German words
  - Each with a human-annotated similarity score
- Cosine similarity as similarity measure for word vectors
- Spearman's rank correlation between cosine similarity of word vectors and their relatedness score?

Training German Word Embeddings: Intrinsic Evaluation

Model	cbow	window	sample	hs	negative	min-count
0	0	8	0	1	10	50
1	1	8	0	1	10	50
2	0	5	0	1	10	50
3	0	15	0	1	10	50
4	0	8	1e-5	1	10	50
5	0	8	1e-3	1	10	50
6	0	8	0	0	10	50
7	0	8	0	1	0	50
8	0	8	0	1	20	50
9	0	8	0	1	10	10
10	0	8	0	1	10	100

Hyper-parameters used for the 11 trained word vector models

Model	0	1	2	3	4	5	6	7	8	9	10
ρ	0.7153	0.5510	0.6933	0.7325	0.7479	0.7335	0.7399	0.7002	0.7103	0.7222	0.7249

Spearman's ρ between the trained word embeddings and Gur350 word pairs human-annotated semantic relatedness

Training German Word Embeddings: Intrinsic Evaluation

First conclusions

- Results in line with recommendations for English embeddings

Training German Word Embeddings: Intrinsic Evaluation

First conclusions

- Results in line with recommendations for English embeddings
  - SG better than CBOW

Training German Word Embeddings: Intrinsic Evaluation

First conclusions

- Results in line with recommendations for English embeddings
  - SG better than CBOW
  - Subsampling most frequent words improves performance

Training German Word Embeddings: Intrinsic Evaluation

First conclusions

- Results in line with recommendations for English embeddings
  - SG better than CBOW
  - Subsampling most frequent words improves performance
  - Large window sizes better than small window sizes

Training German Word Embeddings: Intrinsic Evaluation

First conclusions

- Results in line with recommendations for English embeddings
  - SG better than CBOW
  - Subsampling most frequent words improves performance
  - Large window sizes better than small window sizes
- But... Is this evaluation meaningful for a "real-world task"?

Predictive Sentiment Analysis

Motivation

- Predict user preferences from user textual-feedback
- Check quality of our German word embeddings

Predictive Sentiment Analysis

Motivation

- Predict user preferences from user textual-feedback
- Check quality of our German word embeddings

Predictive Sentiment Analysis

Motivation

- Predict user preferences from user textual-feedback
- Check quality of our German word embeddings

Training corpus: SCARE

- 802,860 German app reviews with star rating from Google Play Store
- 11 app categories, 10-15 apps per category
- Preprocessing identical to German word embedding training
- For classification:
  - Fitness trackers only: 22,118 reviews
  - Rating binarization: True class for ratings with 3 stars or more
  - 26/74 class distribution ratio

Predictive Sentiment Analysis: Approach

Training document embeddings

- Gensim `doc2vec` PV-DBOW implementation
- Training on all SCARE reviews
- Import of word embeddings pretrained on SdeWaC

Predictive Sentiment Analysis: Approach

Training document embeddings

- Gensim `doc2vec` PV-DBOW implementation
- Training on all SCARE reviews
- Import of word embeddings pretrained on SdeWaC

Training classifiers

- Predict positive (4-5 stars) or negative (1-3 stars) ratings
- Learned document embeddings as the only features
- 4 different methods:
  - Logistic Regression (LR)
  - Decision Trees (DR)
  - Random Forests with 101 random trees (RF1)
  - Random Forests with 11 random trees (RF2)

Predictive Sentiment Analysis: Results

Geometric Mean of Negative and Positive Class Accuracy Values of Classifying of Liking of Fitness Tracker Applications. Each model corresponds to a different pre-trained word embedding model

Model	**LR**	**DT**	**RF1**	**RF2**
0	0.791	0.677	0.729	0.696
1	0.766	0.650	0.661	0.645
2	0.789	0.685	0.730	0.695
3	0.788	0.679	0.725	0.686
4	0.784	0.689	0.729	0.691
5	0.791	0.674	0.727	0.697
6	0.788	0.685	0.719	0.693
7	0.790	0.684	0.734	0.699
8	0.783	0.680	0.726	0.693
9	0.799	0.679	0.732	0.693
10	0.789	0.668	0.724	0.693

Conclusions

Semantic analysis (in German) is feasible

- Only using document embeddings as features
- Best models
  - "Just" applied logistic regression
  - $\approx$ 0.8 geometric mean of positive and negative class accuracy

Intrinsic evaluation on a word relatedness dataset: not useful

- No correlation of the score in the intrinsic evaluation with the score in semantic similarity
- In line with (very) recent research showing the unreliability of assessing word embeddings on relatedness datasets
- Extrinsic evaluation recommended instead