Variations to the SkipGram Model

VL Embeddings

Uni Heidelberg

SS 2019

Generalisation of SkipGram to arbitrary contexts

- Neural embeddings so far:
  - linear bag-of-words context (with window size n)

    Die kleine graue Maus frißt den leckeren Käse
- What about other types of contexts?

Generalisation of SkipGram to arbitrary contexts

- Neural embeddings so far:
  - linear bag-of-words context (with window size n)

    Die kleine graue Maus frißt den leckeren Käse

- What about other types of contexts?

Levy and Goldberg (2014):
Dependency-based word embeddings

Starting point: SkipGram

- **Recap**: Skipgram with negative sampling (SGNS)
  - Each word $w \in W$ is associated with a vector $v_w \in \mathbb{R}^d$
  - Each context $c \in C$ is associated with a vector $v_c \in \mathbb{R}^d$
  - W is the word vocabulary
  - C is the context vocabulary
  - d is the embedding dimensionality
- Vector entries are the parameters θ that we want to learn
- Given: dataset D of observed (w, c) pairs in the corpus

- Objective: maximise the probability for seen word-context pairs (w, c) in D and minimise the probability for random word-context pairs in D'

Starting point: SkipGram

- **Recap**: Skipgram with negative sampling (SGNS)
  - Each word $w \in W$ is associated with a vector $v_w \in \mathbb{R}^d$
  - Each context $c \in C$ is associated with a vector $v_c \in \mathbb{R}^d$
  - W is the word vocabulary
  - C is the context vocabulary
  - d is the embedding dimensionality
- Vector entries are the parameters θ that we want to learn
- Given: dataset D of observed (w, c) pairs in the corpus

- SGNS training objective:

$$argmax_{v_w, v_c} \Big(\sum_{(w,c) \in D} log \sigma(v_c \cdot v_w) + \sum_{(w,c) \in D'} log \sigma(-v_c \cdot v_w) \Big)$$

where $\sigma(x) = 1/(1 + e^x)$ sigmoid function

Starting point: SGNS

SGNS

- Observed word-context pairs will end up with similar embeddings
- Context is defined as a bag-of-words window with size n
- Model is unsensitive to position in context window

Starting point: SGNS

SGNS

- Observed word-context pairs will end up with similar embeddings
- Context is defined as a bag-of-words window with size n
- Model is unsensitive to position in context window

Dependency-based embeddings

- Replace bag-of-words context with syntactic context

Dependency-based word embeddings

Australian scientist discovers star with telescope

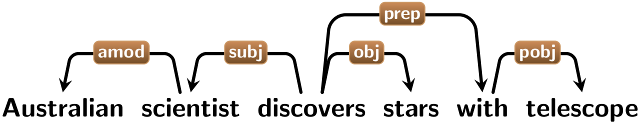
- Which word-context pairs does SGNS extract for discover?
- Which word-context pairs does SGNS extract for star?

Dependency-based word embeddings

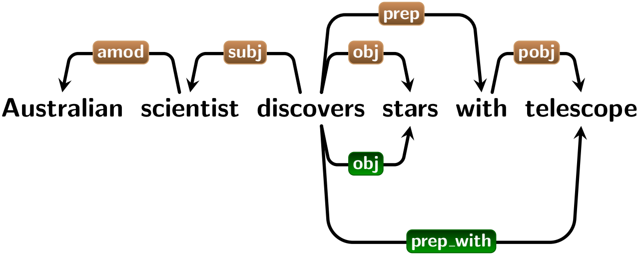
Australian scientist discovers star with telescope

- Which word-context pairs does SGNS extract for discover?
- Which word-context pairs does SGNS extract for star?
- How does the dependency tree for this sentence look like?
- What contexts could a dependency-based model extract?

Dependency-based word embeddings

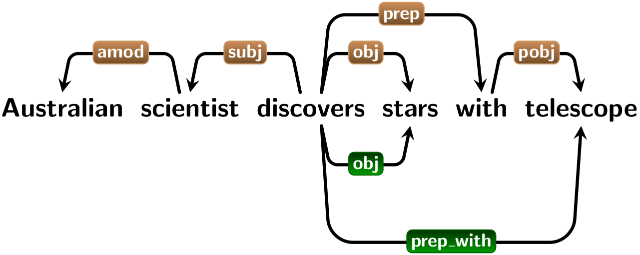


Dependency-based word embeddings



- Collapse preposition relations into single arc (attach PP obj to head of preposition but keep information on prep form).

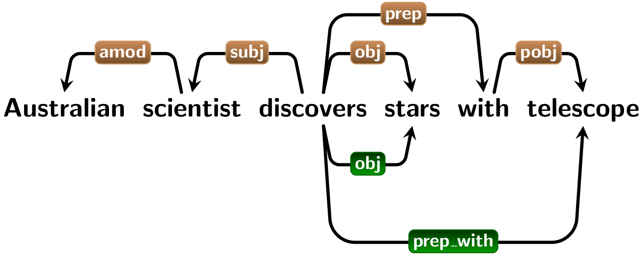
Dependency-based word embeddings



- Collapse preposition relations into single arc (attach PP obj to head of preposition but keep information on prep form).

Word	Contexts

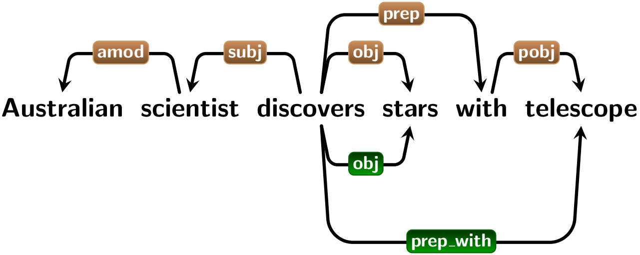
Dependency-based word embeddings



- Collapse preposition relations into single arc (attach PP obj to head of preposition but keep information on prep form).

Word	Contexts
Australian	scientist / amod^{-1}

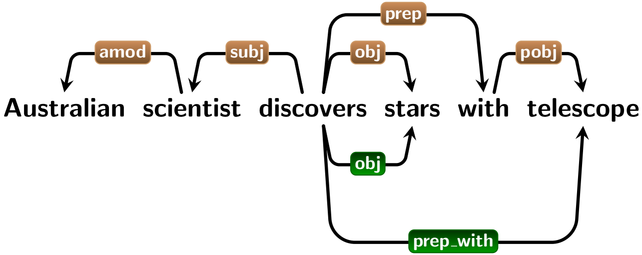
Dependency-based word embeddings



- Collapse preposition relations into single arc (attach PP obj to head of preposition but keep information on prep form).

Word	Contexts
Australian	scientist / amod^{-1}
scientist	australian / amod, discovers / subj^{-1}

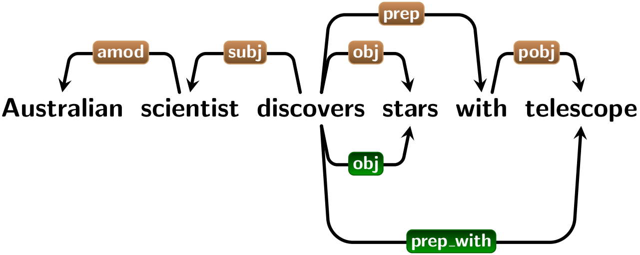
Dependency-based word embeddings



- Collapse preposition relations into single arc (attach PP obj to head of preposition but keep information on prep form).

WORD	CONTEXTS
Australian	scientist / amod^{-1}
scientist	australian / amod, discovers / subj^{-1}
discovers	scientist / subj, star / obj, telescope / prep_with

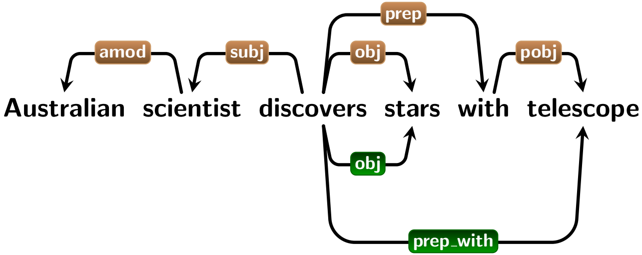
Dependency-based word embeddings



- Collapse preposition relations into single arc (attach PP obj to head of preposition but keep information on prep form).

Word	Contexts
Australian	scientist / amod^{-1}
scientist	australian / amod, discovers / subj^{-1}
discovers	scientist / subj, star / obj, telescope / prep_with
star	discovers / obj^{-1}

Dependency-based word embeddings



- Collapse preposition relations into single arc (attach PP obj to head of preposition but keep information on prep form).

WORD	CONTEXTS
Australian	scientist / amod^{-1}
scientist	australian / amod, discovers / subj^{-1}
discovers	scientist / subj, star / obj, telescope / prep_with
star	discovers / obj^{-1}
telescope	discovers / prep_with^{-1}

Dependency-based word embeddings

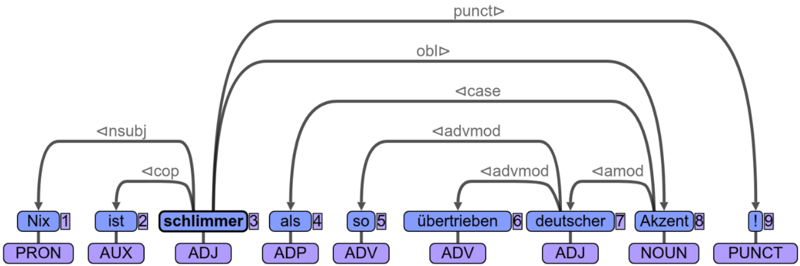
Extract syntactic context

- Parse the corpus
- for a target word w with dependents $m_1, \ldots, m_k$ and a head h

$\Rightarrow$ extract contexts $(m_1, lbl_1), \ldots, (m_k, lbl_k), (h, lbl_h^{-1})$

Dependency-based word embeddings

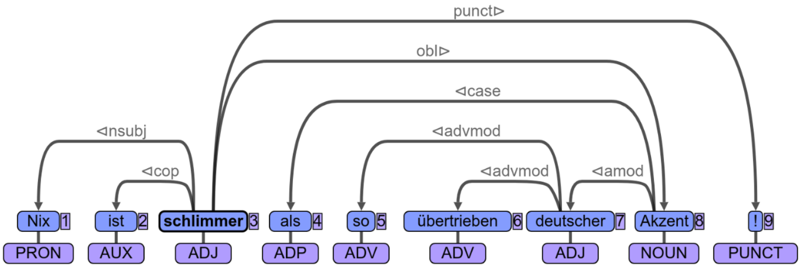
- Given the following tree in Universal Dependencies schema:



- Extract all context words for
  - schlimmer
  - Akzent

Dependency-based word embeddings

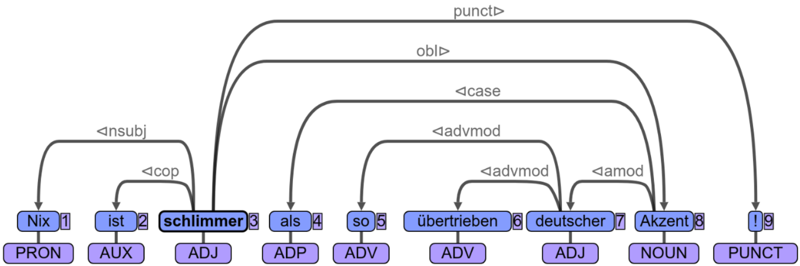
- Given the following tree in Universal Dependencies schema:



- Extract all context words for
  - schlimmer Nix/nsubj, ist/cop, Akzent/obl, !/punct
  - Akzent

Dependency-based word embeddings

- Given the following tree in Universal Dependencies schema:



- Extract all context words for
  - schlimmer — Nix/nsubj, ist/cop, Akzent/obl, !/punct
  - Akzent — deutscher/amod, schlimmer/prep_als^{-1}

Advantages of dependency-based embeddings

- Captures context that is functionally related but far away
- Ignores words that are close by but not related
- Captures general functional relations (e.g. *stars* are objects of *discovery*, *scientists* are subjects of *discovery*)

Advantages of dependency-based embeddings

- Captures context that is functionally related but far away
- Ignores words that are close by but not related
- Captures general functional relations (e.g. *stars* are objects of *discovery*, *scientists* are subjects of *discovery*)

- Hypothesis: Dependency-based embeddings will capture more functional and less topical similarity.

Related work

- Previous work in **distributional semantics**
  - Lin (1998)
  - Padó and Lapata (2007)
  - Baroni and Lenci (2010)
  - ...

Syntax-based semantic space models

Experiments: Settings & Data

Settings

- 3 Training conditions
  - BoW context with size $k = 5$
  - BoW context with size $k = 2$
  - Dependency context
- modified version of SkipGram implementation
- negative samples = 15
- embedding dimensions = 300

Data

- All embeddings trained on English Wikipedia
  - all tokens lower-cased
  - all word-context pairs less frequent than 100 were ignored
- Vocabulary size: 175,000 words
- Over 900,000 distinct syntactic contexts

Qualitative Evaluation

- Manually inspect 5 most similar words (cosine similarity) of a given target word

Findings:

⇒ BoW finds words that associate with *w*
⇒ Deps finds words that behave like *w*

Domain similarity vs. functional similarity

Qualitative Evaluation

Target Word	BoW5	BoW2	Deps
batman	nightwing aquaman catwoman superman manhunter	superman superboy aquaman catwoman batgirl	superman superboy supergirl catwoman aquaman
hogwarts	dumbledore hallows half-blood malfoy snape	evernight sunnydale garderobe blandings collinwood	sunnydale collinwood calarts greendale millfield
turing	nondeterministic non-deterministic computability deterministic finite-state	non-deterministic finite-state nondeterministic buchi primality	pauling hotelling heting lessing hamming
florida	gainesville fla jacksonville tampa lauderdale	fla alabama gainesville tallahassee texas	texas louisiana georgia california carolina
object-oriented	aspect-oriented smalltalk event-driven prolog domain-specific	aspect-oriented event-driven objective-c dataflow 4gl	event-driven domain-specific rule-based data-driven human-centered
dancing	singing dance dances dancers tap-dancing	singing dance dances breakdancing clowning	singing rapping breakdancing miming busking

from Levy & Goldberg (2014)

Qualitative Evaluation

- Hogwards: domain vs semantic type (famous schools)

target word	BoW5	BoW2	Deps
hogwarts	dumbledore	evernight	sunnydale
	hallows	sunnydale	collinwood
	half-blood	garderobe	calarts
	malfoy	blandings	greendale
	snape	collinwood	millfield

Qualitative Evaluation

- Florida: bag-of-words contexts generate meronyms (counties or cities within Florida), while dependency-based contexts provide cohyponyms (other US states)

target word	BoW5	BoW2	Deps
florida	gainesville	fla	texas
	fla	alabama	louisiana
	jacksonville	gainesville	georgia
	tampa	tallahassee	california
	lauderdale	texas	carolina

Qualitative Evaluation

- object-oriented, dancing: dep-based embeddings share a syntactic function (adjectives, gerunds)

target word	BoW5	BoW2	Deps
object-oriented	aspect-oriented	aspect-oriented	event-driven
	smalltalk	event-driven	domain-specific
	event-driven	objective-c	rule-based
	prolog	dataflow	data-driven
	domain-specific	4gl	human-centered

Qualitative Evaluation

- object-oriented, dancing: dep-based embeddings share a syntactic function (adjectives, gerunds)

target word	BoW5	BoW2	Deps
dancing	singing	singing	singing
	dance	dance	rapping
	dances	dances	breakdancing
	dancers	breakdancing	miming
	tap-dancing	clowning	busking

Qualitative Evaluation

- object-oriented, dancing: dep-based embeddings share a syntactic function (adjectives, gerunds)

target word	BoW5	BoW2	Deps
dancing	singing	singing	singing
	dance	dance	rapping
	dances	dances	breakdancing
	dancers	breakdancing	miming
	tap-dancing	clowning	busking

Larger window size → more topicality

Quantitative Evaluation: WordSim353

- Word pairs that show
  - relatedness (topical similarity)
  - similarity (functional similarity)
- Task setup
  - rank the *similar* pairs above the *related* ones
  - ranking according to cosine similarity between embeddings
  - draw recall-precision curve that describes the embedding's affinity towards one subset over another

Quantitative Evaluation: WordSim353

- Word pairs that show
  - relatedness (topical similarity)
  - similarity (functional similarity)
- Task setup
  - rank the *similar* pairs above the *related* ones
  - ranking according to cosine similarity between embeddings
  - draw recall-precision curve that describes the embedding's affinity towards one subset over another

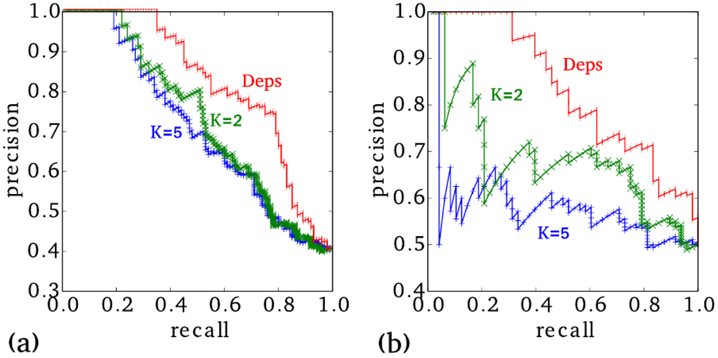
What behaviour would you expect?

Quantitative Evaluation: WordSim353

- Word pairs that show
  - relatedness (topical similarity)
  - similarity (functional similarity)
- Task setup
  - rank the *similar* pairs above the *related* ones
  - ranking according to cosine similarity between embeddings
  - draw recall-precision curve that describes the embedding's affinity towards one subset over another

Expectation: Curve for DEPS > BOW2 > BOW5

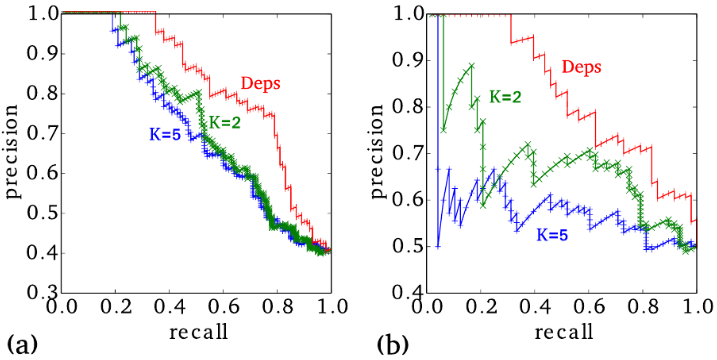
Quantitative Evaluation: WordSim353



from Levy & Goldberg (2014)

- Recall-precision curve when ranking *similar* words above *related* words
  - (a) based on WordSim353 dataset
  - (b) based on Chiarello et al. (1990) dataset (domain vs. function)

Quantitative Evaluation: WordSim353

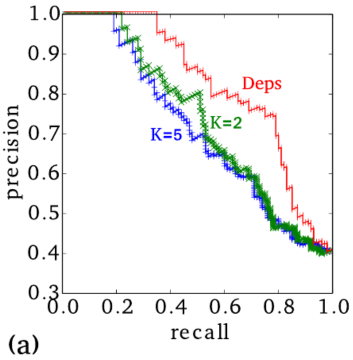


from Levy & Goldberg (2014)

- Recall-precision curve when ranking *similar* words above *related* words
  - (a) based on WordSim353 dataset
  - (b) based on Chiarello et al. (1990) dataset (domain vs. function)

What results would you expect when using dependency-based embeddings for the analogy task?

Quantitative Evaluation: WordSim353



from Levy & Goldberg (2014)

- Recall-precision curve when ranking *similar* words above *related* words
  - (a) based on WordSim353 dataset
  - (b) based on Chiarello et al. (1990) dataset (domain vs. function)

What results would you expect when using dependency-based embeddings for the analogy task? Dependencies worse than BoW for analogies

Insights into the model

- Neural word embeddings are often considered uninterpretable, unlike sparse, count-based distributional representations where each dimension corresponds to a particular known context
  - $\Rightarrow$ not possible to assign a meaning to each dimension

How can we get insights into neural word embeddings?

Insights into the model

- Neural word embeddings are often considered uninterpretable, unlike sparse, count-based distributional representations where each dimension corresponds to a particular known context

  $\Rightarrow$ not possible to assign a meaning to each dimension

How can we get insights into neural word embeddings?

- Examine which contexts are *activated* by a target word
- Model learns to maximise the dot product $v_c \cdot v_w$ for observed word pairs (w, c)
  - Keep context embeddings
  - Which contexts are most activated by a given target word (i.e.: have the highest dot product)

Insights into the model

- List 5 most activated contexts for example words
- Most discriminative syntactic contexts

batman	**hogwarts**	**turing**
superman/conj^{-1}	students/prep_at^{-1}	machine/nn^{-1}
spider-man/conj^{-1}	educated/prep_at^{-1}	test/nn^{-1}
superman/conj	student/prep_at^{-1}	theorem/poss^{-1}
spider-man/conj	stay/prep_at^{-1}	machines/nn^{-1}
robin/conj	learned/prep_at^{-1}	tests/nn^{-1}
florida	**object-oriented**	**dancing**
marlins/nn^{-1}	programming/amod^{-1}	dancing/conj
beach/appos^{-1}	language/amod^{-1}	dancing/conj^{-1}
jacksonville/appos^{-1}	framework/amod^{-1}	singing/conj^{-1}
tampa/appos^{-1}	interface/amod^{-1}	singing/conj
florida/conj^{-1}	software/amod^{-1}	ballroom/nn

from Levy & Goldberg (2014)

Generalisation of SGNS

Sum-up

- Generalisation of linear bag-of-words context to arbitrary contexts
  - here: dependency-based contexts
- Depending on the context, the model learns different properties from the same data

Generalisation of SGNS

Sum-up

- Generalisation of linear bag-of-words context to arbitrary contexts
  - here: dependency-based contexts
- Depending on the context, the model learns different properties from the same data

Dependency-based embeddings

- are less topical and exhibit more functional similarity than the original skipgram embeddings

Generalisation of SGNS

Sum-up

- Generalisation of linear bag-of-words context to arbitrary contexts
  - here: dependency-based contexts
- Depending on the context, the model learns different properties from the same data

Dependency-based embeddings

- are less topical and exhibit more functional similarity than the original skipgram embeddings

What other contexts are possible?

FastText – Background

- Mikolov et al. 2013: Distributed Representations of words and phrases and their compositionality
  - Representation of words in vector space
  - Drawbacks:
    - no sentence representations
    - does not exploit morphology
      (different representations for disaster / disastrous)

FastText – Motivation

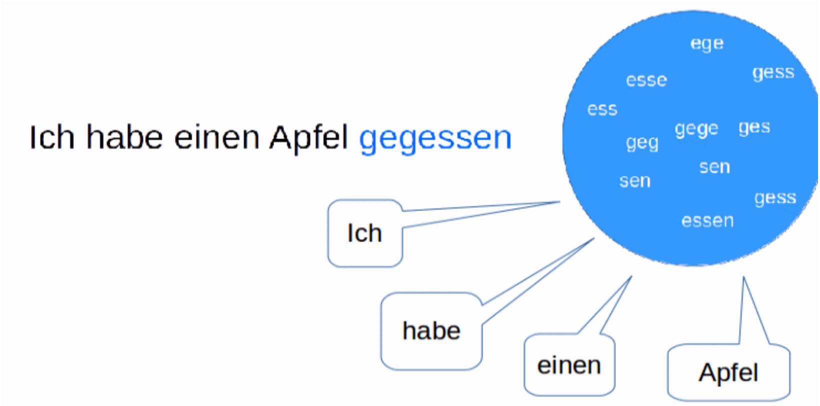
- Better representations for morphological variants of same word
- Better representations for rare/unseen words

$\Rightarrow$ Train word representations with character-level features

FastText – Motivation

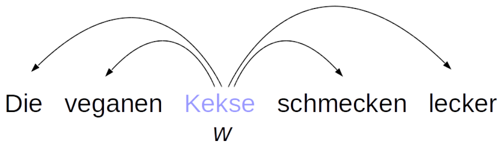
- Better representations for morphological variants of same word
- Better representations for rare/unseen words

$\Rightarrow$ Train word representations with character-level features



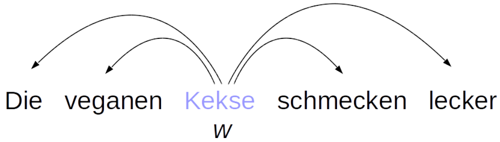
- Use character ngrams to predict surrounding context

Recap: SkipGram



Kekse $\rightarrow$ Die
Kekse $\rightarrow$ veganen
Kekse $\rightarrow$ schmecken
Kekse $\rightarrow$ lecker

Recap: SkipGram



Kekse $\rightarrow$ Die
Kekse $\rightarrow$ veganen
Kekse $\rightarrow$ schmecken
Kekse $\rightarrow$ lecker

- Model probability of a context word given a word

representation for word w: v_w
representation for context word c: v_c

$$p(c|w) = \frac{e^{v_w^\top v_c}}{\sum_{k=1}^{K} e^{v_w^\top v_k}}$$

- Word vectors $v_w \in \mathbb{R}^d$

Softmax

Recap: SkipGram

Die veganen Kekse schmecken lecker

w (Kekse)

Kekse $\rightarrow$ Die
Kekse $\rightarrow$ veganen
Kekse $\rightarrow$ schmecken
Kekse $\rightarrow$ lecker

- Model probability of a context word given a word

representation for word w: v_w
representation for context word c: v_c

$$p(c|w) = \frac{e^{v_w^\top v_c}}{\sum_{k=1}^{K} e^{v_w^\top v_k}}$$

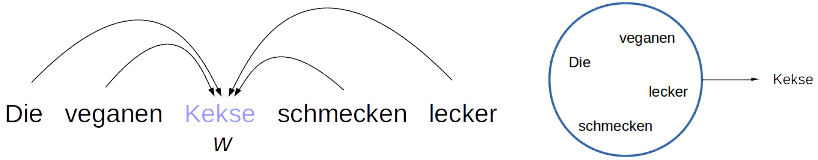
Softmax

- Word vectors $v_w \in \mathbb{R}^d$
- Softmax computationally expensive

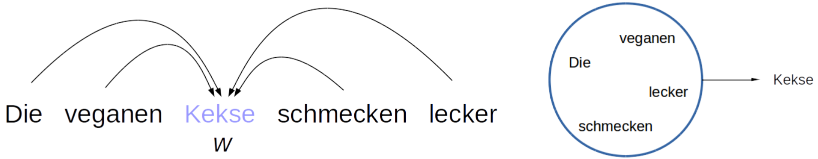
$\Rightarrow$ use approximations:

- Hierarachical softmax
- Negative sampling $log(1 + e^{-v_{w_t}^\top v_c}) + \sum_{n \in N_c} log(1 + e^{v_{w_t}^\top v_n})$

Recap: CBOW


Die veganen Kekse schmecken lecker
w
veganen
Die
lecker
schmecken
Kekse

Recap: CBOW



- Model probability of a word given the context

representation for context C: h_c
representation for word w: v_w

$$p(w|C) = \frac{e^{h_c^\top v_w}}{\sum_{k=1}^{K} e^{h_c^\top v_k}}$$

- Continuous bag of words (sum of the words in the context) $h_c = \sum_{c \in C} v_c$

FastText

- As in SkipGram: model probability of a context word c given a word w

representation for word w: h_w
representation for word c: v_c

$$p(c|w) = \frac{e^{h_w^\top v_c}}{\sum_{k=1}^{K} e^{h_w^\top v_k}}$$

FastText

- As in SkipGram: model probability of a context word c given a word w

representation for word w: h_w
representation for word c: v_c

$$p(c|w) = \frac{e^{h_w^\top v_c}}{\sum_{k=1}^{K} e^{h_w^\top v_k}}$$

- Representation of a word w computed based on ngrams: all ngrams with length l where $3 \leq l \leq 6$ *and* word form

FastText

- As in SkipGram: model probability of a context word c given a word w

representation for word w: h_w
representation for word c: v_c

$$p(c|w) = \frac{e^{h_w^\top v_c}}{\sum_{k=1}^{K} e^{h_w^\top v_k}}$$

- Representation of a word w computed based on ngrams: all ngrams with length l where $3 \leq l \leq 6$ *and* word form

$$h_w = \sum_{\substack{g_l \in w \\ 3 \leq l \leq 6}} v_g$$

	zwi			
ker		wink		
	zwink		$+$	zwinkert
kert		inke		
	ert	...		
	char ngrams			word form

Advantages of FastText

Out-of-Vocabulary (OOV) words

- Ngram representations are shared across words
  $\Rightarrow$ more reliable representations for rare words
- We now can build vectors for unseen words:

Advantages of FastText

Out-of-Vocabulary (OOV) words

- Ngram representations are shared across words
  $\Rightarrow$ more reliable representations for rare words
- We now can build vectors for unseen words:

$$h_w = \sum_{g \in w} v_g$$

	ver		
ker		wink	
	zwink		+ ~~verzwinkert~~
kert		inke	
	ert	...	

char ngrams   word form

FastText Training

- Training with Stochastic Gradient Descent
- Minimise negative log-likelihood
- Set ngram length $= 0 \Rightarrow$

FastText Training

- Training with Stochastic Gradient Descent
- Minimise negative log-likelihood
- Set ngram length $= 0 \Rightarrow$ SkipGram with negative sampling

FastText Training

- Training with Stochastic Gradient Descent
- Minimise negative log-likelihood
- Set ngram length $= 0 \Rightarrow$ SkipGram with negative sampling

- Evaluation – model parameters:
  - 300 dimensions
  - sample 5 negative examples per word
  - context window size c, uniformly sample c between 1 and 5
  - subsample frequent words with threshold 10^{-4}
  - discard all words that occur < 5 times in the corpus
  - learning rate 0.05

FastText Training

- Training with Stochastic Gradient Descent
- Minimise negative log-likelihood
- Set ngram length $= 0 \Rightarrow$ SkipGram with negative sampling

- Evaluation – model parameters:
  - 300 dimensions
  - sample 5 negative examples per word
  - context window size c, uniformly sample c between 1 and 5
  - subsample frequent words with threshold 10^{-4}
  - discard all words that occur < 5 times in the corpus
  - learning rate 0.05

- Training speed:
  - Model is around $1.5\times$ slower than SkipGram

Word Similarity Evaluation

- Given: pair of words w_1, w_2
- Compare cosine similarity for w_1, w_2 against human judgements

  $s(w_1, w_2) = \frac{x_{w1}^\top x_{w_q}}{||x_{w_1}|| \; ||x_{w_2}||}$
- Spearman's rank correlation

		SG	CBOW	FT*	FT
AR	WS353	51	52	54	**55**
DE	GUR350	61	62	64	**70**
	GUR65	78	78	**81**	**81**
	ZG222	35	38	41	**44**
EN	RW	43	43	46	**47**
	WS353	72	**73**	71	71
ES	WS353	57	58	58	**59**
FR	RG65	70	69	**75**	**75**
RO	WS353	48	52	51	**54**
RU	HJ	59	60	60	**66**

FT* uses null vector for unknowns

Word Similarity Evaluation

- Given: pair of words w_1, w_2
- Compare cosine similarity for w_1, w_2 against human judgements

$$s(w_1, w_2) = \frac{x_{w1}^\top x_{w_q}}{||x_{w_1}||\ ||x_{w_2}||}$$

- Spearman's rank correlation

		SG	**CBOW**	**FT***	**FT**
AR	WS353	51	52	54	**55**
DE	GUR350	61	62	64	**70**
	GUR65	78	78	**81**	**81**
	ZG222	35	38	41	**44**
EN	RW	43	43	46	**47**
	WS353	72	**73**	71	71
ES	WS353	57	58	58	**59**
FR	RG65	70	69	**75**	**75**
RO	WS353	48	52	51	**54**
RU	HJ	59	60	60	**66**

FT* uses null vector for unknowns

Works particularly well for datasets with rare words and for morphologically rich languages

Word Analogy Evaluation

- Paris $\rightarrow$ France; Rom $\rightarrow$?
  - Predict the analogy
  - Evaluate using accuracy

What results would you expect?

Word Analogy Evaluation

- Paris $\rightarrow$ France; Rom $\rightarrow$?
  - Predict the analogy
  - Evaluate using accuracy

		SG	CBOW	FT
CS	Semantic	**25.7**	27.6	27.5
	Syntactic	52.8	55.0	**77.8**
DE	Semantic	66.5	**66.8**	62.3
	Syntactic	44.5	45.0	**56.4**
EN	Semantic	**78.5**	78.2	77.8
	Syntactic	70.1	69.9	**74.9**
IT	Semantic	52.3	**54.7**	52.3
	Syntactic	51.5	51.8	**62.7**

Word Analogy Evaluation

- Paris → France; Rom → ?
  - Predict the analogy
  - Evaluate using accuracy

		SG	CBOW	FT
CS	Semantic	**25.7**	27.6	27.5
	Syntactic	52.8	55.0	**77.8**
DE	Semantic	66.5	**66.8**	62.3
	Syntactic	44.5	45.0	**56.4**
EN	Semantic	**78.5**	78.2	77.8
	Syntactic	70.1	69.9	**74.9**
IT	Semantic	52.3	**54.7**	52.3
	Syntactic	51.5	51.8	**62.7**

- Works well for syntactic analogies, especially for morphologically rich languages (CS, DE)

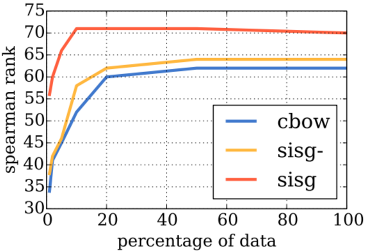
  groß → größer; hoch → ?

Effect of training data size

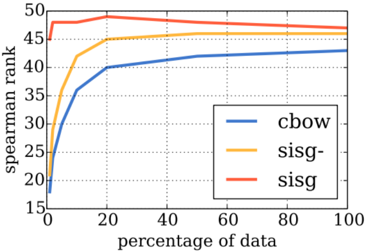
- FastText works well for rare and unknown words
- Hypothesis: FastText is also better in settings where we do not have a lot of training data.
- Test: train CBOW and FastText on subsets of Wikipedia (1, 2, 5, 10, 20, 50%)

Effect of training data size

- FastText works well for rare and unknown words
- Hypothesis: FastText is also better in settings where we do not have a lot of training data.
- Test: train CBOW and FastText on subsets of Wikipedia (1, 2, 5, 10, 20, 50%)



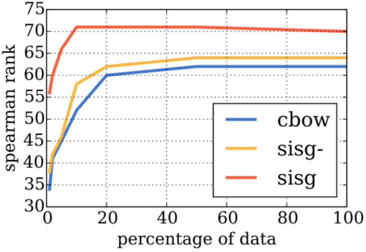
(a) DE-GUR350



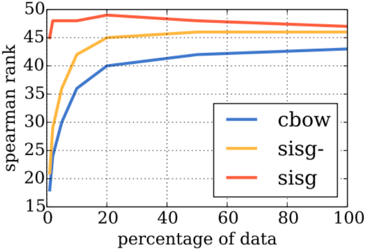
(b) EN-RW

Effect of training data size

- FastText works well for rare and unknown words
- Hypothesis: FastText is also better in settings where we do not have a lot of training data.
- Test: train CBOW and FastText on subsets of Wikipedia (1, 2, 5, 10, 20, 50%)



(a) DE-GUR350

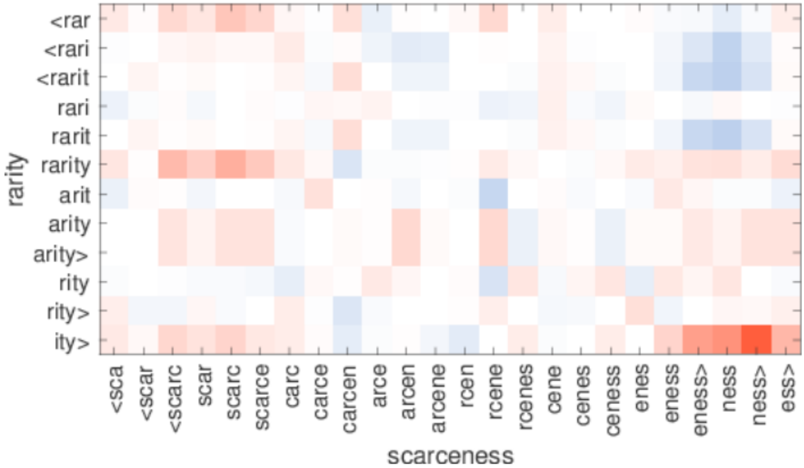


(b) EN-RW

- Adding more data does not always improve results

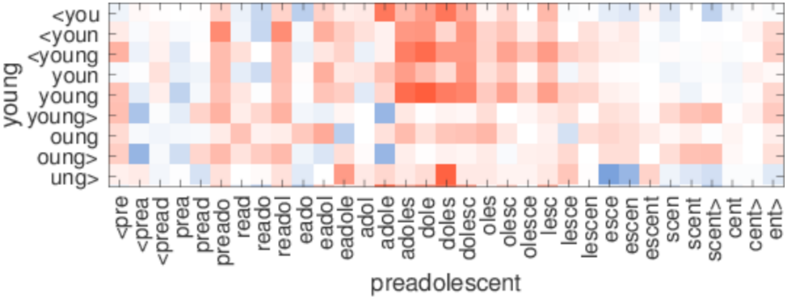
Word similarity evaluation for unknown words

- Train on 1% of EN Wikipedia
- Report cosine similarity for ngrams of word pairs where one word is unknown



Word similarity evaluation for unknown words

- Train on 1% of EN Wikipedia
- Report cosine similarity for ngrams of word pairs where one word is unknown



FastText – Sum-up

- Extension of the SGNS model that represents each word by the sum of its subword representations
- For ngram length=0 $\Rightarrow$ same as SGNS
- Fast to train, good results for smaller training data sizes
- Superior performance especially for rare and unknown words and for syntactic analogies

References

- Marco Baroni and Alessandro Lenci (2010): Distributional memory: A general framework for corpus-based semantics. Computational Linguistics, 36(4):673–721.
- Yoav Goldberg and Omer Levy (2014): word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. arXiv preprint arXiv:1402.3722.
- Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In Proceedings of the 36th Annual Meeting of the Association for Computational Lin- guistics and 17th International Conference on Com- putational Linguistics - Volume 2, ACL '98, pages 768–774, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean (2013): Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States, pages 3111–3119.
- Sebastian Padó and Mirella Lapata (2007): Dependency-based construction of semantic space models. Computational Linguistics, 33(2):161–199.
- Omer Levy and Yoav Goldberg (2014): Dependency-based word embeddings. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 302–308, Baltimore, Maryland, USA