Embedding Words and Senses Together via Joint Knowledge-Enhanced Training

Manchini et al.

Damyana Gateva

Saarland University

February 1, 2018

Overview

Outline

Motivation

- Word sense disambiguation using knowledge from semantic networks
- Joint learning of embeddings of words and senses

He withdrew money from the bank.

Motivation

- Previous approaches
  - Unsupervised sense embeddings

Motivation

- Previous approaches
  - Unsupervised sense embeddings
    - + learn senses only from text corpora
    - - induced senses are not interpretable or mappable to lexical resources
    - - infrequent senses difficult to discriminate

Motivation

- Previous approaches
  - Unsupervised sense embeddings
    - + learn senses only from text corpora
    - - induced senses are not interpretable or mappable to lexical resources
    - - infrequent senses difficult to discriminate
  - Knowledge-based sense embeddings

Motivation

- Previous approaches
  - Unsupervised sense embeddings
    - \+ learn senses only from text corpora
    - \- induced senses are not interpretable or mappable to lexical resources
    - \- infrequent senses difficult to discriminate
  - Knowledge-based sense embeddings
    - \+ use predefined senses from semantic networks
    - \- a training step in addition to word embeddings
    - \- do not solve the meaning conflation issue properly
    - \- infrequent senses difficult to discriminate

Motivation

- SW2V: Senses and Words to Vectors
  - + exploits knowledge from both text corpora and semantic networks
  - + jointly training of words and sense embeddings
  - + uses one training step
  - + represents word and sense embeddings in the same vector space
  - + can be applied to different predictive models
  - + is scalable for large semantic networks and text corpora
  - + captures infrequent senses

Corpus and semantic network

- corpus: UMBC 300M-words corpus and Wikipedia
- semantic network: BabelNet
  - over 350M semantic connections
  - integrates Wikipedia and WordNet

Outline

Method

Input: corpus + semantic network

1. use a semantic network to link associated senses in context
   $\rightarrow$ shallow word-sense connectivity algorithm

2. use a neural network with linked word and sense embeddings
   $\rightarrow$ joint update

Shallow word-sense connectivity algorithm

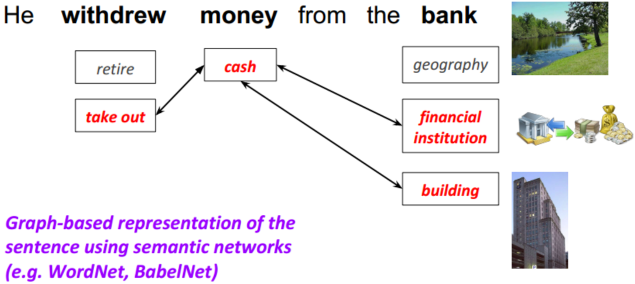
1) gather S_T: all candidate synsets of the words* in the text

Shallow word-sense connectivity algorithm

1) gather S_T: all candidate synsets of the words* in the text
2) for each candidate s $\in S_T$ calculate number of synsets connected with the semantic network

Shallow word-sense connectivity algorithm

1) gather S_T: all candidate synsets of the words* in the text
2) for each candidate $s \in S_T$ calculate number of synsets connected with the semantic network



Shallow word-sense connectivity algorithm

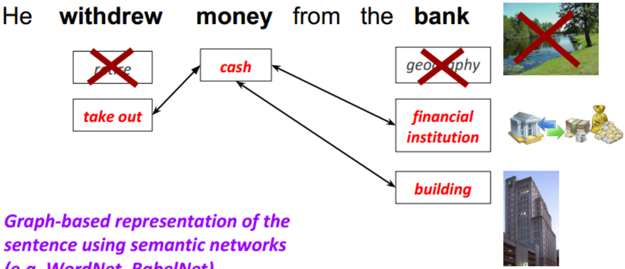
1) gather S_T: all candidate synsets of the words* in the text
2) for each candidate s $\in S_T$ calculate number of synsets connected with the semantic network
3) retain connections above a threshold θ

Shallow word-sense connectivity algorithm

1) gather S_T: all candidate synsets of the words* in the text
2) for each candidate $s \in S_T$ calculate number of synsets connected with the semantic network
3) retain connections above a threshold θ
4) associate each word* with top candidate synsets according to their number of connections in context $\longrightarrow$ semantic network graph (S, E)

Shallow word-sense connectivity algorithm

Semantic network graph (S, E)



Graph-based representation of the sentence using semantic networks (e.g. WordNet, BabelNet)

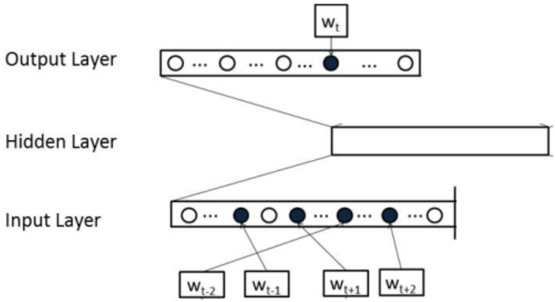
Model

- extension of the word2vec CBOW architecture: + **senses**
- word2vec
  - feed forward neural network
  - **CBOW**: predicting the current word w_t using its context
  - also applicable to Skip-Gram

Model

word2vec CBOW: predicting the current word w_t using its context

$$E = -\log(p(w_t | W^t))$$



Model

- extension of the word2vec CBOW architecture: + **senses**
- word2vec
  - feed forward neural network
  - **CBOW**: predicting the current word w_t using its context
  - also applicable to Skip-Gram
- SW2V: predicting the current word w_t + **its set of associated senses** S_t

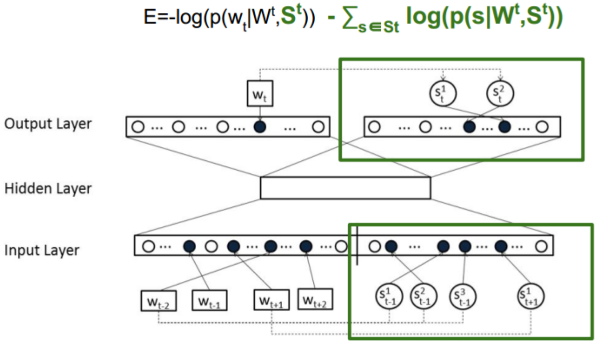
Model

- extension of the word2vec CBOW architecture: + **senses**
- word2vec
  - feed forward neural network
  - **CBOW**: predicting the current word w_t using its context
  - also applicable to Skip-Gram
- SW2V: predicting the current word w_t + **its set of associated senses** S_t

A word is a surface form of a underlying sense
$\Rightarrow$ "updating the embedding of a word should produce a consequent update to the embedding representing that particular sense, and vice-versa"

Model

SW2V: predicting the current word w_t + **its set of associated senses** S_t



Words and associated senses used both as input and output

Model parameters

SW2V: predicting the current word w_t + **its set of associated senses** S_t

- vector dimensionality: 300
- window size: 8
- normalization: hierarchical softmax

Model variants

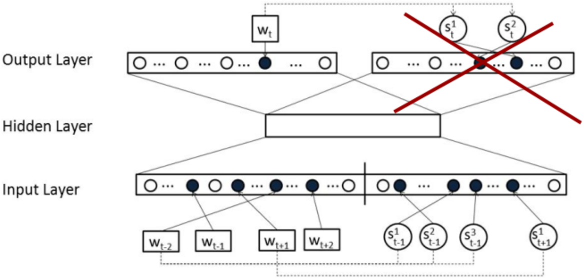
Input and output layer alternatives $\rightarrow$ calculation of the hidden state and contribution to the loss function

- both words and senses
- only words
- only senses

Model variants

Output layer alternatives: only words

$$E=-\log(p(w_t|W^t,S^t)) - \cancel{\sum_{s\in S^t} \log(p(s|W^t,S^t))}$$

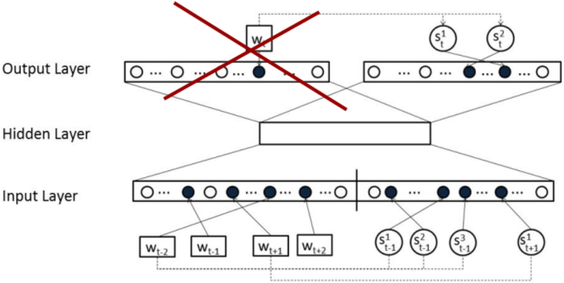


The architecture does not try to predict **senses** $\Rightarrow$ No loss contribution from them

Model variants

Output layer alternatives: only senses

$$E=-\cancel{\log(p(w_t|W^t,S^t))} - \sum_{s\in S^t} \log(p(s|W^t,S^t))$$

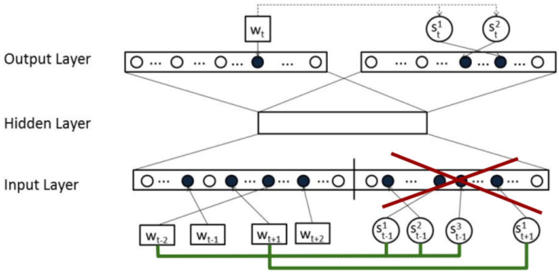


The architecture does not try to predict **words**. $\Rightarrow$ No loss contribution from them.

Model variants

Input layer alternatives: only words

$$E=-\log(p(w_t|W^t,\cancel{S^t})) - \sum_{s\in S_t} \log(p(s|W^t,\cancel{S^t}))$$

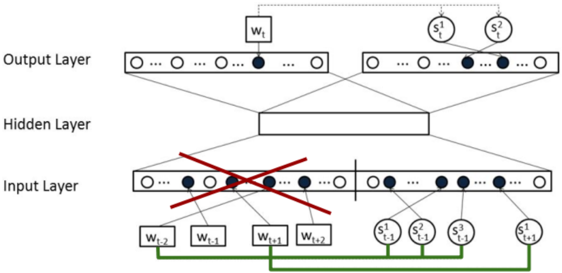


Senses do not contribute to the hidden layer. During backpropagation **sense embeddings** receive the **same gradient of the word they are associated with**.

Model variants

Input layer alternatives: only senses

$$E=-\log(p(w_t|\cancel{W^t},S^t)) - \sum_{s\in St} \log(p(s|\cancel{W^t},S^t))$$



Words do not contribute to the hidden layer. During backpropagation **word embeddings** receive the **same gradient of the senses they are associated with**.

Analysis of model configuration

- Tests on word similarity with each of the 9 configurations
- Best configuration:
  - Input layer: only senses
  - Output layer: both words and senses

Analysis of model configuration

- Tests on word similarity with each of the 9 configurations
- Best configuration:
  - Input layer: only senses
  - Output layer: both words and senses
- Intuition: “Co-occurrence information gets duplicated if both words and senses are included in the input layer”

Analysis of model configuration

Best configuration:

		Output											
		Words				Senses				Both			
		WS-Sim		RG-65		WS-Sim		RG-65		WS-Sim		RG-65	
		r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
Input	**Words**	0.49	0.48	0.65	0.66	0.56	0.56	0.67	0.67	0.54	0.53	0.66	0.65
	Senses	0.69	0.69	0.70	0.71	0.69	0.70	0.70	**0.74**	**0.72**	**0.71**	**0.71**	**0.74**
	Both	0.60	0.65	0.67	0.70	0.62	0.65	0.66	0.67	0.65	**0.71**	0.68	0.70

Pearson r and Spearman ρ correlation performance

Outline

Evaluation: shallow word-sense connectivity algorithm

- input: pre-disambiguated text
- baseline: Babelfly - state-of-the-art graph-based disambiguation and entity linking system
  (* only instances above the default confidence threshold disambiguated)
- results:

  - better correlation results
  - 10 times faster than Babelfly
  - more robust by associating words with more than one sense

	WS-Sim		RG-65	
	r	ρ	r	ρ
Shallow	**0.72**	**0.71**	**0.71**	**0.74**
Babelfy	0.65	0.63	0.69	0.70
Babelfy*	0.63	0.61	0.65	0.64

Pearson r and Spearman ρ correlation performance

Evaluation: Model

- Best configuration used on all experiments
- Experiments on:
  - Word similarity
  - Sense clustering
  - Word and sense interconnectivity
- Measure of word similarity: cosine similarity
- Measure of sense similarity: closest sense strategy
  $sim(w_1, w_2) = max_{s \in S_{w_1}, s' \in S_{w_2}} cos(\overrightarrow{s_1}, \overrightarrow{s_2})$

Model results

- Word similarity

	System	Corpus	SimLex-999 r	SimLex-999 ρ	MEN r	MEN ρ
Senses	$SW2V_{BN}$	UMBC	**0.49**	**0.47**	0.75	0.75
	$SW2V_{WN}$	UMBC	0.46	0.45	**0.76**	**0.76**
	AutoExtend	UMBC	0.47	0.45	0.74	0.75
	AutoExtend	Google-News	0.46	0.46	0.68	0.70
	$SW2V_{BN}$	Wikipedia	0.47	0.43	0.71	0.73
	$SW2V_{WN}$	Wikipedia	0.47	0.43	0.71	0.72
	SensEmbed	Wikipedia	0.43	0.39	0.65	0.70
	Chen et al. (2014)	Wikipedia	0.46	0.43	0.62	0.62
Words	Word2vec	UMBC	0.39	0.39	0.75	0.75
	$Retrofitting_{BN}$	UMBC	0.47	0.46	0.75	**0.76**
	$Retrofitting_{WN}$	UMBC	0.47	0.46	**0.76**	**0.76**
	Word2vec	Wikipedia	0.39	0.38	0.71	0.72
	$Retrofitting_{BN}$	Wikipedia	0.35	0.32	0.66	0.66
	$Retrofitting_{WN}$	Wikipedia	0.47	0.44	0.73	0.73

Pearson r and Spearman ρ correlation performance on the SimLex-999 and MEN word similarity datasets

Model results

- Sense clustering
- Binary classification task - a pair is a cluster above a threshold γ

	Accuracy	F-Measure
SW2V	**87.8**	**63.9**
SensEmbed	82.7	40.3
NASARI	87.0	62.5
Multi-SVM	85.5	-
Mono-SVM	83.5	-
Baseline	17.5	29.8

Accuracy and F-score of different systems on the SemEval Wikipedia sense clustering dataset, BabelNet only as lexical resource

Model results

- Word and sense interconnectivity
  - Intuition: the most common sense (MCS) should be close to the word embedding

$$MCS(w) = argmax_{s \in S_w} cos(\vec{w}, \vec{s})$$

	SemEval-07	SemEval-13
SW2V	**39.9**	**54.0**
AutoExtend	17.6	31.0
Baseline	24.8	34.9

F-score of different MCS strategies

Model results

- Word and sense interconnectivity

company^2_n (military unit)		*school^7_n (group of fish)*	
AutoExtend	**SW2V**	**AutoExtend**	**SW2V**
company^9_n	battalion^1_n	school	schools^7_n
company	battalion	school^4_n	sharks^1_n
company^8_n	regiment^1_n	school^6_n	sharks
company^6_n	detachment^4_n	school^1_v	shoals^3_n
company^7_n	platoon^1_n	school^3_n	fish^1_n
company^1_v	brigade^1_n	elementary	dolphins^1_n
firm	regiment	schools	pods^3_n
business^1_n	corps^1_n	elementary^3_a	eels
firm^2_n	brigade	school^5_n	dolphins
company^1_n	platoon	elementary^1_a	whales^2_n

Outline

Conclusion

- Joint vector space for words and sense embeddings: semantically coherent vector space
- One training phase
- Better results on all 3 tasks
- Able to disambiguate also less frequent senses
- Quick and scalable

References

Mancini, Massimiliano et al. (2016a). "Embedding Words and Senses Together via Joint Knowledge-Enhanced Training". In: *arXiv preprint arXiv:1612.02703*.

– (2016b). *Embedding Words and Senses Together via Joint Knowledge-Enhanced Training, Tutorial*. URL: https://de.slideshare.net/aclanthology/massimiliano-mancini-2017-embeddings-words-and-senses-together-via-joint-knowledgeenhanced-training (visited on 01/25/2018).