Making Sense of Word Embeddings



Maria Pelevina¹, Nikolay Arefiev², Chris Biemann¹, Alexander Panchenko¹ ¹ TU Darmstadt, Germany ² Moscow State University, Russia



illustration

Introduction

We present a simple yet effective approach for learning word sense embeddings. In contrast to existing techniques, which either directly learn sense representations from corpora or rely on sense inventories from lexical resources, our approach can induce a sense inventory from existing word embeddings via clustering of egonetworks of related words. An integrated WSD mechanism enables labeling of words in context with learned sense vectors, which gives rise to downstream applications. Experiments show that the performance of our method is comparable to state-of-theart unsupervised WSD systems.

Word Sense Induction



Learning Sense Embeddings from Word Embeddings



Sense Vectors

Schema of the word sense embeddings learning method.

	TWSI	JBT	w2v
table (furniture)	counter, console, bench, dinner table, dining table, desk, sur- face, bar, board	chair, room, desk, pulpit, couch, furniture, fireplace, bench, door, box, railing, tray	tray, bottom, bucket, basket, cup, pile, bracket, pot, cue, plate, jar, platter, ladder
table (data)	chart, list, index, graph, columned list, tabulation, standings, diagram, ranking	procedure, system, datum, pro- cess, mechanism, tool, method, database, calculation, scheme	diagram, brackets, stack, list, parenthesis, playfield, drop- down, cube, hash, results, tab
table (negotiations)	surface, counter, console, bar- gaining table, platform, nego- tiable, machine plate, level		
table (geo)	level, plateau, plain, flatland,		

Visualization of the ego-network of the word "table" with the "furniture" and the "data" sense clusters.

Algorithm 1: Word sense induction.input : T – word similarity graph, N –
ego-network size, n – ego-network
connectivity, k – minimum cluster sizeoutput: for each term $t \in T$, a clustering S_t of its
N most similar termsforeach $t \in T$ do $V \leftarrow N$ most similar terms of t from T
 $G \leftarrow$ graph with V as nodes and no edges Eforeach $v \in V$ do $V' \leftarrow n$ most similar terms of v from T
foreach $v' \in V'$ do| if $v' \in V$ then add edge (v, v') to E
end

table (geo)

ographical level, water table, ge-

Word sense clusters from inventories derived from the Wikipedia corpus via crowdsourcing (TWSI), JoBimText (JBT) and word embeddings (w2v).

Word Sense Disambiguation with Sense Embeddings

Context representation based	Similarity- and probability-bas	sed disambiguation in context:
on k context or word vectors:	$s^* = \operatorname*{argmax}_{i} sim(\mathbf{s}_i, C)$	$C(t) = \arg\max_{i} \frac{\mathbf{c}_{w} \cdot \mathbf{s}_{i}}{\ \bar{\mathbf{c}}_{w}\ \cdot \ \mathbf{s}_{i}\ }$
$\bar{\mathbf{c}}_c = k^{-1} \sum_{i=1}^n \operatorname{vec}_c(c_i)$	$s^* = rg\max_i P(C \mathbf{s}_i)$	$= \arg\max_{i} \frac{1}{1 + e^{-\bar{\mathbf{c}}_c \cdot \mathbf{s}_i}}$
$ar{\mathbf{c}}_w = k^{-1} \sum_{i=1}^k vec_w(c_i)$ s	Sense vector: $b_i = rac{\sum_{k=1}^n \gamma_i(w_k) vec_w(w_k)}{\sum_{k=1}^n \gamma_i(w_k)}$	Filtering the k context words: $\max_{i} f(\mathbf{s}_{i}, c_{j}) - \min_{i} f(\mathbf{s}_{i}, c_{j})$

 $\begin{array}{l} \textbf{end} \\ S_t \leftarrow \texttt{ChineseWhispers}(G) \\ S_t \leftarrow \{s \in S_t : |s| \geq k\} \\ \textbf{end} \end{array}$

Vector	Nearest Neighbours				
table	 tray, bottom, diagram, bucket, brackets, stack, basket, list, parenthesis, cup, trays, pile, play- field, bracket, pot, drop-down, cue, plate 				
table#0	<pre>leftmost#0, column#1, randomly#0, tableau#1, top-left0, indent#1, bracket#3, pointer#0, footer#1, cursor#1, diagram#0, grid#0</pre>				
table#1	<pre>pile#1, stool#1, tray#0, basket#0, bowl#1, bucket#0, box#0, cage#0, saucer#3, mirror#1, birdcage#0, hole#0, pan#1, lid#0</pre>				

Neighbours of the word "table" and its senses.

Results: WSD Evaluation on the TWSI and the SemEval 2013 Task 13 Datasets

				Supervised Evaluation			luation	Clustering Evaluation		
	a much manaishtad much	Trucialitad aims Trucialita	d aires Elter (n-2) ZNEC Norman have d	Model		Jacc. Ind.	Tau	WNDCG	F.NMI	F.B-Cubed
\square random \square mean prob. \square weighted prob. \square weighted si		weighted sim. weighte	n. Weighted sim filter ($p=2$) \boxtimes MFS \boxtimes upper bound	Baselines	One sense for all	0.171	0.627	0.302	0.000	0.631
1.00	Full TWSI Dataset	1 00	Sense-balanced TWSI Dataset		One sense per instance	0.000	0.953	0.000	0.072	0.000
1.00		1.00			Most Eraguant Sansa (MES)	0 570	0 5 9 2	0 421		



Performance of our method trained on the Wikipedia corpus on the full (on the left) and on the sense-balanced (on the right) TWSI dataset.

DEG Deutsche Forschungsgemeinschaft

Most Frequent Sense (MFS)	0.579	0.385	0.451	_	
AI-KU (add1000)	0.176	0.609	0.205	0.033	0.317
AI-KU	0.176	0.619	0.393	0.066	0.382
AI-KU (remove5-add1000)	0.228	0.654	0.330	0.040	0.463
Unimelb (5p)	0.198	0.623	0.374	0.056	0.475
Unimelb (50k)	0.198	0.633	0.384	0.060	0.494
UoS (#WN senses)	0.171	0.600	0.298	0.046	0.186
UoS (top-3)	0.220	0.637	0.370	0.044	0.451
La Sapienza (1)	0.131	0.544	0.332	_	_
La Sapienza (2)	0.131	0.535	0.394	_	-
AdaGram, $\alpha = 0.05$, 100 dim. vectors	0.274	0.644	0.318	0.058	0.470
w2v – weighted – sim. – filter ($p = 2$)	0.197	0.615	0.291	0.011	0.615
w2v – weighted – sim. – filter $(p = 2)$: nouns	0.179	0.626	0.304	0.011	0.623
JBT – weighted – sim. – filter $(p = 2)$	0.205	0.624	0.291	0.017	0.598
JBT – weighted – sim. – filter $(p = 2)$: nouns	0.198	0.643	0.310	0.031	0.595
TWSI – weighted – sim. – filter $(p = 2)$: nouns	0.215	0.651	0.318	0.030	0.573
	Most Frequent Sense (MFS) AI-KU (add1000) AI-KU AI-KU (remove5-add1000) Unimelb (5p) Unimelb (50k) UoS (#WN senses) UoS (top-3) La Sapienza (1) La Sapienza (2) AdaGram, $\alpha = 0.05$, 100 dim. vectors w2v - weighted - sim filter ($p = 2$) w2v - weighted - sim filter ($p = 2$): nouns JBT - weighted - sim filter ($p = 2$): nouns TWSI - weighted - sim filter ($p = 2$): nouns	Most Frequent Sense (MFS) 0.379 AI-KU (add1000) 0.176 AI-KU 0.176 AI-KU (remove5-add1000) 0.228 Unimelb (5p) 0.198 Unimelb (50k) 0.198 UoS (#WN senses) 0.171 UoS (top-3) 0.220 La Sapienza (1) 0.131 La Sapienza (2) 0.131 AdaGram, $\alpha = 0.05$, 100 dim. vectors 0.274 w2v - weighted - sim filter ($p = 2$) 0.197 w2v - weighted - sim filter ($p = 2$) 0.179 JBT - weighted - sim filter ($p = 2$) 0.205 JBT - weighted - sim filter ($p = 2$): nouns 0.198 TWSI - weighted - sim filter ($p = 2$): nouns 0.198	Most Frequent Sense (MFS) 0.379 0.383 AI-KU (add1000) 0.176 0.609 AI-KU 0.176 0.619 AI-KU (remove5-add1000) 0.228 0.654 Unimelb (5p) 0.198 0.623 Unimelb (50k) 0.198 0.633 UoS (#WN senses) 0.171 0.600 UoS (top-3) 0.220 0.637 La Sapienza (1) 0.131 0.544 La Sapienza (2) 0.131 0.535 AdaGram, $\alpha = 0.05$, 100 dim. vectors 0.274 0.644 w2v - weighted - sim filter ($p = 2$): 0.197 0.615 w2v - weighted - sim filter ($p = 2$): 0.197 0.626 JBT - weighted - sim filter ($p = 2$): 0.198 0.643 TWSI - weighted - sim filter ($p = 2$): 0.198 0.643	Most Frequent Sense (MFS) 0.379 0.383 0.431 AI-KU (add1000) 0.176 0.609 0.205 AI-KU 0.176 0.619 0.393 AI-KU (remove5-add1000) 0.228 0.654 0.330 Unimelb (5p) 0.198 0.623 0.374 Unimelb (5k) 0.198 0.633 0.384 UoS (#WN senses) 0.171 0.600 0.298 UoS (top-3) 0.220 0.637 0.370 La Sapienza (1) 0.131 0.544 0.332 La Sapienza (2) 0.131 0.535 0.394 AdaGram, $\alpha = 0.05$, 100 dim. vectors 0.274 0.644 0.318 w2v - weighted - sim filter ($p = 2$) 0.197 0.615 0.291 w2v - weighted - sim filter ($p = 2$): nouns 0.179 0.626 0.304 JBT - weighted - sim filter ($p = 2$): nouns 0.198 0.643 0.310 TWSI - weighted - sim filter ($p = 2$): nouns 0.215 0.651 0.318	Most Frequent Sense (MFS) 0.379 0.385 0.431 $-$ AI-KU (add1000) 0.176 0.609 0.205 0.033 AI-KU 0.176 0.619 0.393 0.066 AI-KU (remove5-add1000) 0.228 0.654 0.330 0.040 Unimelb (5p) 0.198 0.623 0.374 0.056 Unimelb (50k) 0.198 0.633 0.384 0.060 UoS (#WN senses) 0.171 0.600 0.298 0.046 UoS (top-3) 0.220 0.637 0.370 0.044 La Sapienza (1) 0.131 0.544 0.332 $-$ La Sapienza (2) 0.131 0.535 0.394 $-$ AdaGram, $\alpha = 0.05$, 100 dim. vectors 0.274 0.644 0.318 0.058 w2v - weighted - sim filter ($p = 2$): nouns 0.179 0.615 0.291 0.011 JBT - weighted - sim filter ($p = 2$): nouns 0.198 0.643 0.310 0.031 TWSI - weighted - sim filter ($p = 2$): nouns 0.205 0.651 0.318 0.030

The best configurations of our method on the SemEval 2013 Task 13 dataset. All systems were trained on the ukWaC corpus

Code & Data: https://github.com/tudarmstadt-lt/sensegram