Distributional Semantic Models
Diving Deeper

LOT Winterschool 2019, Day 3

Antske Fokkens

Recap

- Distributional semantic models represent word meaning through vectors, or embeddings
- Embeddings reflect the contexts a word occurs in:
  - By counting contexts (PPMI model, SVD)
  - By applying machine learning (inspired) approaches

Evaluating Semantic Models

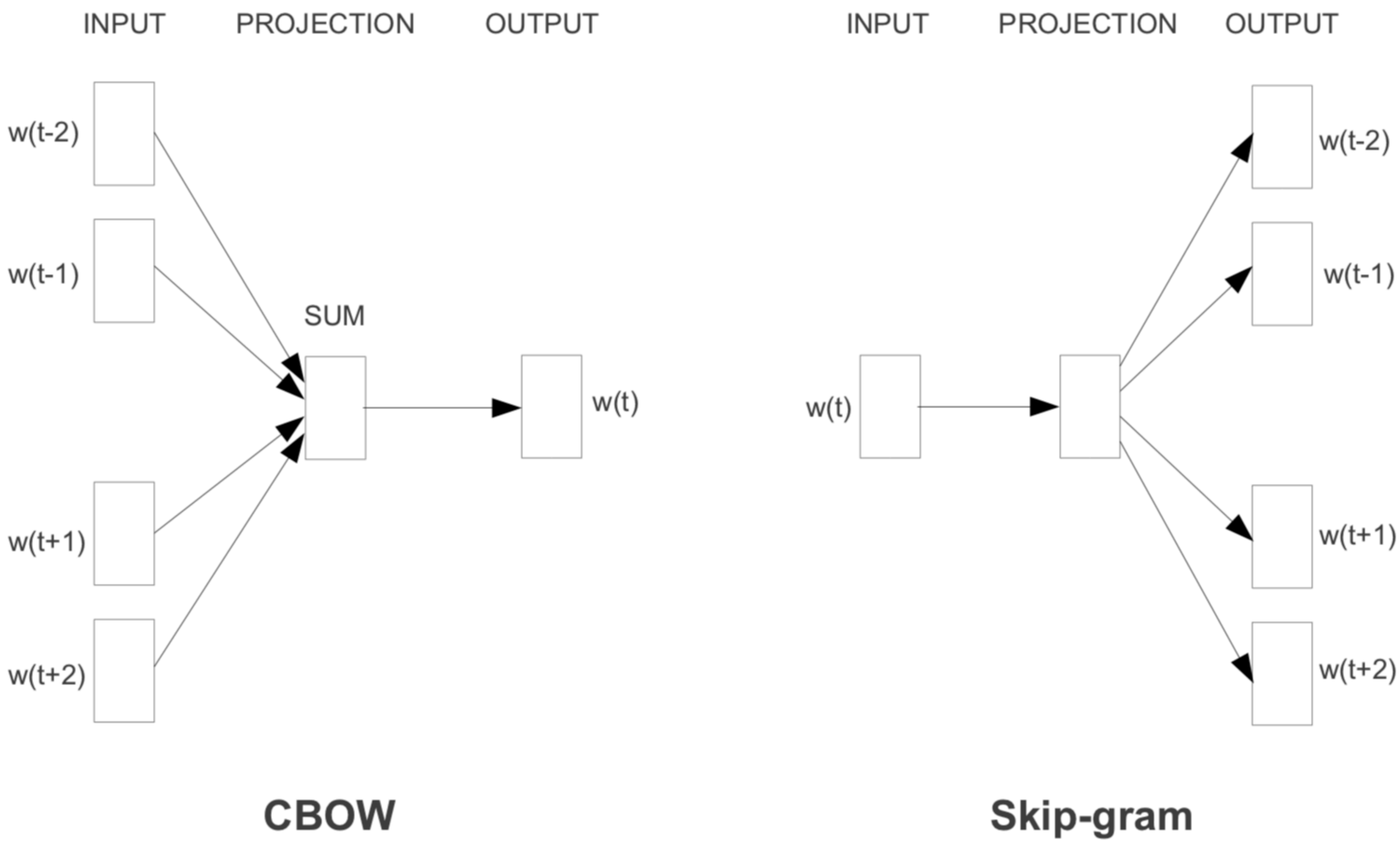
- Intrinsic evaluation:

  Do they provide good representations of meaning?

- Extrinsic evaluation:

  Are they useful for analyzing natural language?

word2vec



Mikolov et al. (2013)

Tasks (examples)

- Language model
- Lemmatizing & pos-tagging
- Dependency parsing
- Word-sense disambiguation
- Semantic role labeling
- Sentiment & opinion mining
- Named Entity Recognition & Classification
- Textual entailment
- Coreference resolution
- Machine translation

Features

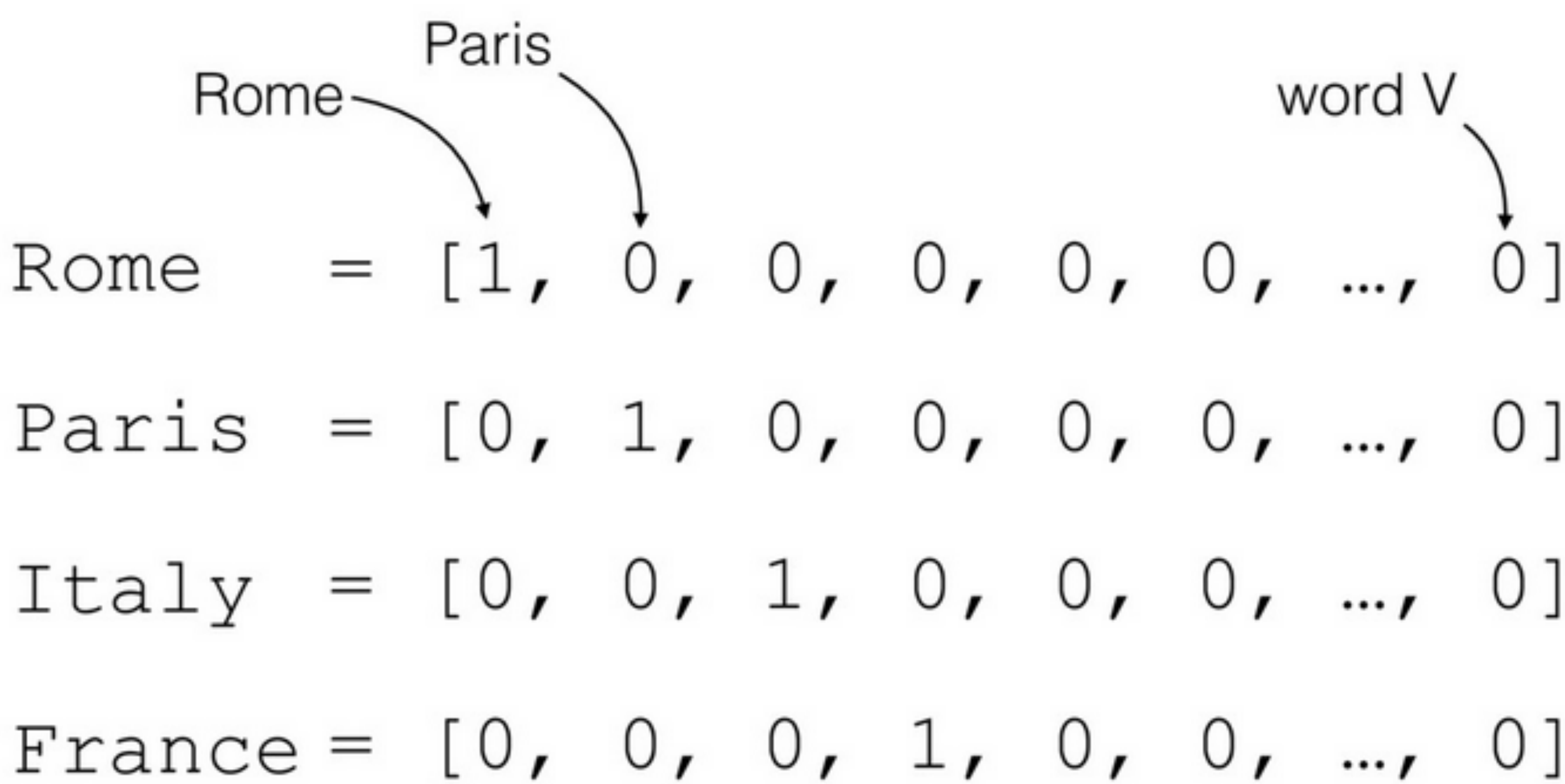
- Common features (for many tasks):
  - POS-tag
  - Word
  - Lemma
  - ngrams

Highly Informative for many tasks!!!

- More advanced:
  - Chunks
  - Syntactic dependencies
  - Word sense

Feature representation

- Basic, old-school: one-hot vector:



from Shaffy (2017)

Distributional Semantic Models

- Can provide high-density representations with less dimension
- Provide similar representations for words with similar surface behavior
- Capture a range of semantic & syntactic properties

Evaluating Semantic Models

- Intrinsic evaluation:

  Do they provide good representations of meaning?

- Extrinsic evaluation:

  Are they useful for analyzing natural language?

Intrinsic Evaluation

- Ranked similarity & relatedness pairs
- Analogy sets

Similarity

- Evaluation for ``general purpose” models that capture semantic similarity

- Assumption:
  => *attributional similarity*: the more attributes that are shared between two concepts, the more similar contexts they occur in
  => *taxonomic similarity*: concepts with high attributional similarity are also taxonomically similar (synonyms, antonyms, co-hyponyms, hyper- and hyponyms)

- Evaluation set-up: can the model identify which word pairs are semantically similar and which are not?

Similarity Tasks

- General procedure:
  - humans indicate how semantically similar two words are:
    - word pairs are rated on a scale
    - humans indicate which out of two word pairs is more semantically similar
  - average rating by multiple annotators leads to score per word pair
  - word pairs are ranked according to their similarity

Dataset

- **WS-353** (Finkelstein et al. 2001): 353 pairs ranked for similarity & relatedness on a scale
    - WS-353-sim: subsection with just similarity or low score
    - WS-353—rel: subsection capturing other forms of relatedness
- **MEN** (Bruni et al. 2012): 3,000 pairs ranked for similarity & relatedness by having humans select the more related pair out of two pairs
- **SimLex-999** (Hill et al. 2015): 999 pairs annotated for similarity only: rated on a scale of 0-6 looking at 7 pairs simultaneously.
- **Radinsky** (Radinsky et al. 2011): 280 pairs of words occurring in the New York times and DBpedia with varying PMI scores. The general approach follows WS-353.
- **Luong** rare words (Luong et al. 2013): at least one of the two words in the pair is rare (5-10, 10-100, 100-1,000, 1,000-10,000 occurrences in wikipedia), filtered using WordNet.

Evaluating on Similarity

- Rank word-pairs by distributional semantic model:
  - the smaller the angle between two vectors, the higher their similarity
- Compare ranking by semantic model to human ranking using Spearman *rho*

Spearman rho

- Calculation: $\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$
- *d* = difference between ranking by model & human
- n = number of samples in the dataset
- (In case of ties in the ranking: assign the mean to all pairs)

Analogy test sets

- Can distributional semantic models capture analogy?
  - Paris:France ~ Rome:Italy
  - queen:king ~ woman:man
  - talk:talked ~ bend:bent
  - man:men ~ pencil:pencils
  - strong:stronger ~ sweet:sweeter

Toy example

Semantic relations via analogies - a toy example

King - man + woman ≈ queen

0	0	1	1
1	1	0	0
1	0	0	1

	female	male	royal
woman	1	0	0
queen	1	0	1
man	0	1	0
king	0	1	1

from Sommerauer

Analogy test

- $V_{king} - V_{man} + V_{woman} \approx V_{queen}$
- $V_{Rome} - V_{Italy} + V_{France} \approx V_{Paris}$
- $V_{stronger} - V_{strong} + V_{sweet} \approx V_{sweeter}$

Analogy test

Must be closest vector to the outcome of the sum

- $V_{king} - V_{man} + V_{woman} \approx \boxed{V_{queen}}$
- $V_{Rome} - V_{Italy} + V_{France} \approx \boxed{V_{Paris}}$
- $V_{stronger} - V_{strong} + V_{sweet} \approx \boxed{V_{sweeter}}$

Analogy test

- $V_{king} - V_{man} + V_{woman} \approx \boxed{V_{queen}}$
- $V_{Rome} - V_{Italy} + V_{France} \approx \boxed{V_{Paris}}$
- $V_{stronger} - V_{strong} + V_{sweet} \approx \boxed{V_{sweeter}}$

Must be closest vector to the outcome of the sum

...excluding all vectors left of the equation

What works best?

We don’t know…

Method	WordSim Similarity	WordSim Relatedness	Bruni et al. MEN	Radinsky et al. M. Turk	Luong et al. Rare Words	Hill et al. SimLex	Google Add / Mul	MSR Add / Mul
PPMI	.755	**.688**	.745	**.686**	.423	.354	.553 / **.629**	.289 / .413
SVD	**.784**	.672	**.777**	.625	**.514**	.402	.547 / .587	.402 / .457
SGNS	.773	.623	.723	.676	.431	**.423**	.599 / .625	.514 / .546
GloVe	.667	.506	.685	.599	.372	.389	.539 / .563	.503 / **.559**
CBOW	.766	.613	.757	.663	.480	.412	.547 / .591	.557 / **.598**

Table 3: Performance of each method across different tasks using word2vec's recommended configuration: win = 2; dyn = with; sub = dirty; neg = 5; cds = 0.75; w+c = only w; eig = 0.0. CBOW is presented for comparison.

Levy et al. (2015)

Method	WordSim Similarity	WordSim Relatedness	Bruni et al. MEN	Radinsky et al. M. Turk	Luong et al. Rare Words	Hill et al. SimLex	Google Add / Mul	MSR Add / Mul
PPMI	.755	**.697**	.745	.686	.462	.393	.553 / .679	.306 / .535
SVD	**.793**	.691	**.778**	.666	**.514**	.432	.554 / .591	.408 / .468
SGNS	**.793**	.685	.774	**.693**	.470	**.438**	.676 / **.688**	.618 / **.645**
GloVe	.725	.604	.729	.632	.403	.398	.569 / .596	.533 / .580

Table 4: Performance of each method across different tasks using the best configuration for that method and task combination, assuming win = 2.

Levy et al. (2015)

Discussing intrinsic evaluation

- Do you think these evaluation methods have problems? If so, what are they?

- How can these datasets be used? If at all?

Criticism from literature

- Similarity (Gladkova & Drozd 2016, among others):
  - Determining which pair is more similar (*money,dollar*) vs (*tiger,mammal*) is difficult: is the difference in score meaningful?
  - Who are the annotators (on mechanical Turk)?

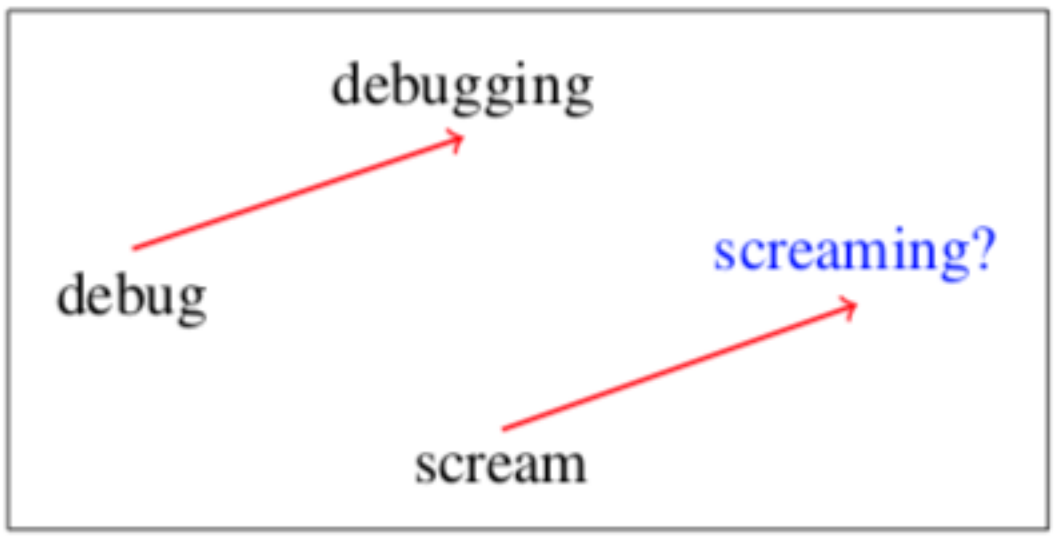
Alternative validation Schnabel et al (2015)

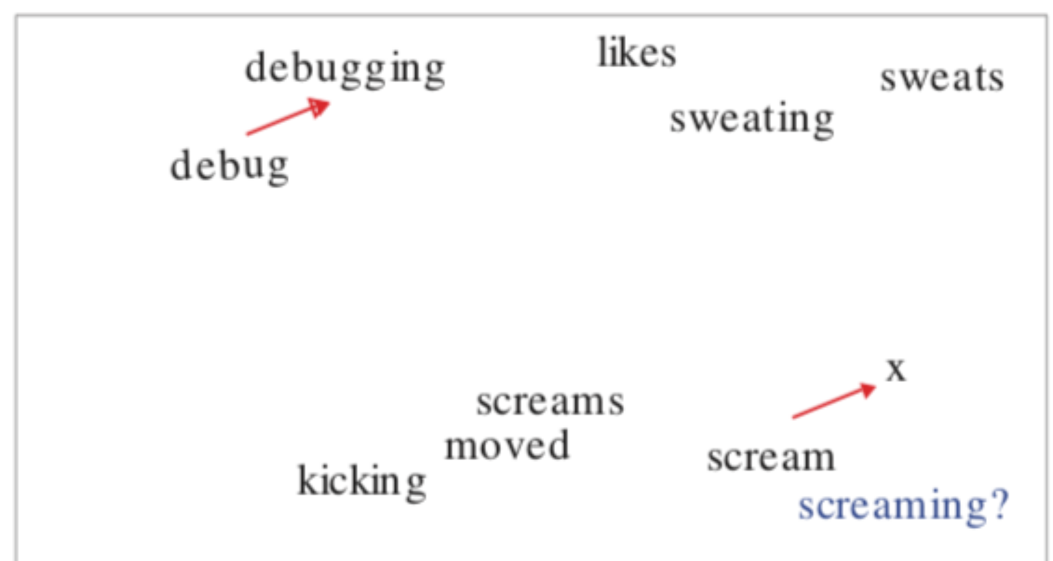
- Identifying the top candidate:
    - present a word with 6 terms from its k-nearest neighbors
    - let annotators pick the most similar term
- Identifying the intruder:
    - present a word with its k-nearest neighbors + a randomly selected word
    - let annotators pick the intruder

Criticism from literature

- Analogy:
  - Linzen (2016): if *a* and *a** are close, *a* - *a** + *b* will be very close to *b*
  - Gladkova et al. (2016): overall results are biased because of overrepresentation of specific types of analogies

Linzen (2016)


debugging
debug
screaming?
scream


debugging
likes
sweats
sweating
debug
x
screams
moved
scream
kicking
screaming?

Linzen (2016)

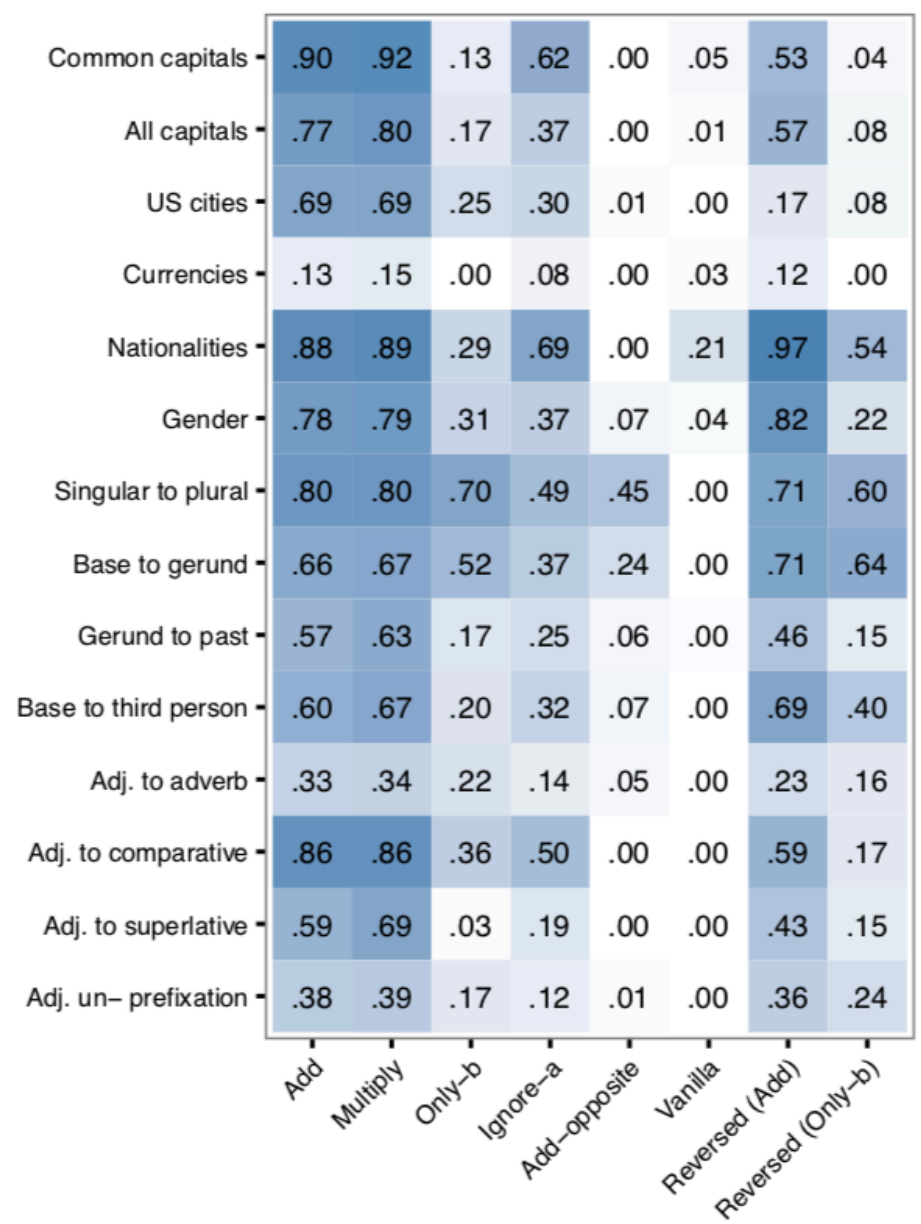
- Vanilla: $$x* = \underset{x'}{\operatorname{argmax}}\ cos(x', a*-a+b)$$
- Add: $$x* = \underset{x' \notin \{a, a*, b\}}{\operatorname{argmax}}\ cos(x', a*-a+b)$$
- Only-B: $$x* = \underset{x' \notin \{a, a*, b\}}{\operatorname{argmax}}\ cos(x', b)$$

Linzen (2016)

- Ignore-A: $x* = \underset{x' \notin \{a,a*,b\}}{\text{argmax}} \; cos(x', a*+b)$
- Add-opposite: $x* = \underset{x' \notin \{a,a*,b\}}{\text{argmax}} \; cos(x', -(a*-a)+b)$
- Reverse (add): $x* = \underset{x' \notin \{a,a*,b*\}}{\text{argmax}} \; cos(x', a-a*+b*)$
- Reverse (B-only)

Linzen (2016)

- Outcome:



References

- Bruni, E., Boleda, G., Baroni, M., & Tran, N. K. (2012, July). Distributional semantics in technicolor. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1 (pp. 136-145). Association for Computational Linguistics.
- Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G. and Ruppin, E., 2002. Placing search in context: The concept revisited. *ACM Transactions on information systems*, *20*(1), pp.116-131.
- Gladkova, Anna, and Aleksandr Drozd. "Intrinsic evaluations of word embeddings: What can we do better?." In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pp. 36-42. 2016.
- Gladkova, Anna, Aleksandr Drozd, and Satoshi Matsuoka. "Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't." In *Proceedings of the NAACL Student Research Workshop*, pp. 8-15. 2016.
- Hill, F., Reichart, R. and Korhonen, A., 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, *41*(4), pp.665-695.
- Levy, Omer, Yoav Goldberg, and Ido Dagan. "Improving distributional similarity with lessons learned from word embeddings." *Transactions of the Association for Computational Linguistics* 3 (2015): 211-225.
- Linzen, T. (2016) "Issues in evaluating semantic spaces using word analogies." *arXiv preprint arXiv:1606.07736*
- Luong, Minh-Thang, Richard Socher, and Christopher D. Manning. 2013. Better word representations with recursive neural networks for morphology. CoNLL-2013, page 104, Sofia.
- Mikolov, T., K. Chen, G. Corrado and J. Dean (2013) Efficient estimation of word representations in vector space. https://arxiv.org/pdf/1301.3781.pdf
- Radinsky, K., Agichtein, E., Gabrilovich, E. and Markovitch, S., 2011, March. A word at a time: computing word relatedness using temporal semantic analysis. In *Proceedings of the 20th international conference on World wide web* (pp. 337-346). ACM.
- Schnabel, Tobias, Igor Labutov, David Mimno, and Thorsten Joachims. "Evaluation methods for unsupervised word embeddings." In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 298-307. 2015.
- Shaffy, Athif (2017) Vector Representation of Text for Machine Learning. https://medium.com/@athif.shaffy/one-hot-encoding-of-text-b69124bef0a7
- Sommerauer, Pia. What is in a word embedding vector?