

Distributional Semantic Models Applications & Evaluations

LOT School, Day 2
Antske Fokkens

Sources

- Baroni & Boleda. Distributional Semantic Models
<https://www.cs.utexas.edu/~mooney/cs388/slides/dist-sem-intro-NLP-class-UT.pdf>
- Goldberg, Yoav (2015) [A primer on neural network models for Natural Language Processing](#)
- 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>
- Shaffy, Athif (2017) Vector Representation of Text for Machine Learning. <https://medium.com/@athif.shaffy/one-hot-encoding-of-text-b69124bef0a7>
- Pia Sommerauer. What is in a word embedding vector?
- All reported sources were accessed between January 14-16 2019

Recap

What do we know about X?

Whereas traditional politicians offer visitors X, the Reform of Heisei serves black coffee.

The river Neckinger, “the colour of strong X”, flowed round Jacob’s Island.

X comes from the leaves that have been withered and dried immediately after picking

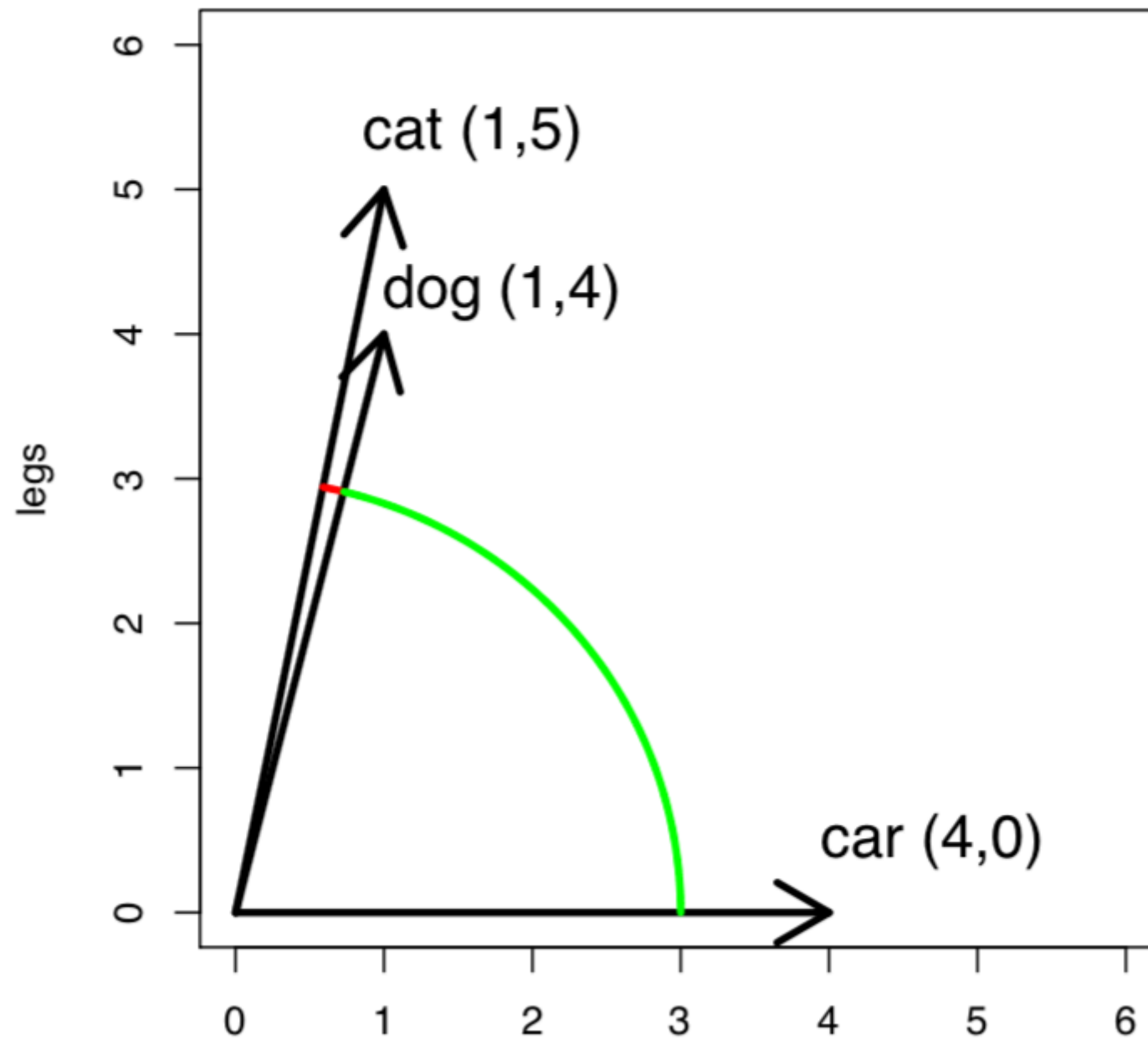
It is a large leaf X with a very delicate flavor

It may be black or X flavored with jasmine flowers, is very fragrant and is always drunk without milk

- a similar category as black coffee:
a (hot) beverage?
- different degrees of strength: mixed/drawn/brewed
- color can be used to describe a river: transparent, blue, green, brown tone
- made from dried leaves
- can have delicate flavor; probably variations in flavor exist
- there is something similar that is black

adapted from Sommerauer

Vector representation



from Baroni & Boleda

Semantic models

- represent meaning as vectors capturing the context in which a word occurs

	leash	walk	run	owner	pet	bark
dog	3	5	2	5	3	2
cat	0	3	3	2	3	0
lion	0	3	2	0	1	0
light	0	0	0	0	0	0
bark	1	0	0	2	1	0
car	0	0	1	3	0	0

Creating Embeddings

1. Preprocess data
2. Select contexts
3. Count contexts & transform counts
or learn to predict them
=> Vector representations of meaning

Selecting context

- What impact do you think context selection has?
 - window-size
 - syntactic restrictions
 - syntactic/pos-tag encoding
 - filtering low frequency terms
 - filtering high frequency terms

How?

- Count models: PPMI, SVD
- Predict models: word2vec, ELMO software packages

Count models: PPMI

- $PPMI = \text{argmax}(0, \log(\frac{P(w1, w2)}{P(w1)P(w2)}))$

$$\log\left(\frac{P(w1, w2)}{P(w_1)P(w_2)}\right)$$

Probability of w1 & w2 occurring together in corpus

Probability of w1 occurring with w2 by chance

PPMI

- High dimensional (size of vocabulary)
- Low density (zeros for all context-words occurring less than by chance)
- Relatively high impact of low frequency words

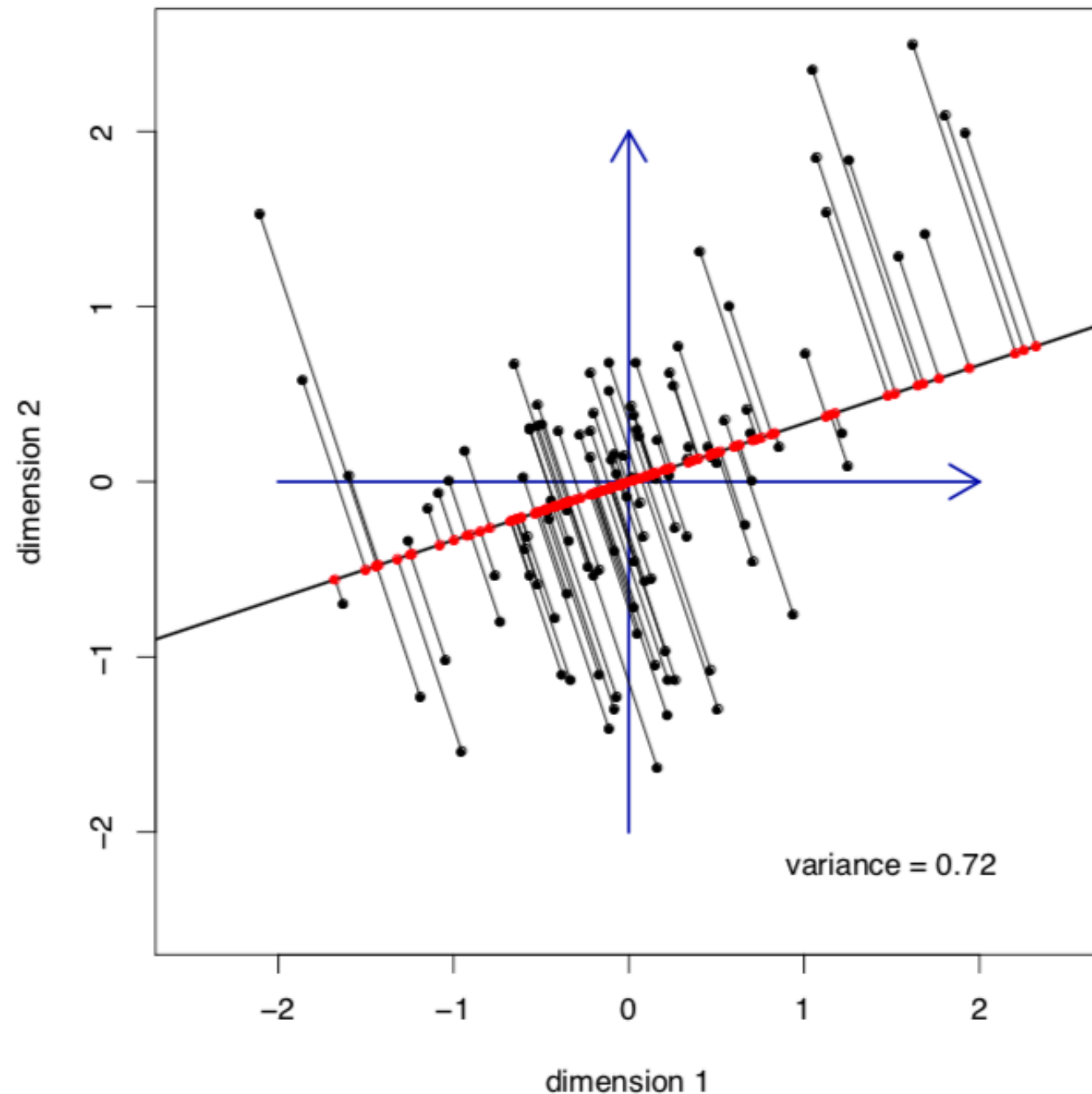
Singular Value Decomposition (SVD)

- Method to reduce the number of dimensions:

given a $m \times n$ matrix, construct a $m \times k$ matrix,
where $k \ll n$

- Uses linear algebra to reduce the number of dimensions,
preserving most of the variance of the original matrix

Capturing most variance



from Baroni & Boleda

Reducing dimensions

- Vocabulary matrix A is decomposed in $U\Sigma V^T$
 - Columns of U and V are ordered according to highest associated eigenvalue
 - Diagonal values of Σ are ordered starting with highest (root of) eigenvalues
- => Using the first d rows of U , the first d columns of V^T and $d \times d$ rows and columns of Σ guarantees that we end up with those values that provide the highest variance.

Result

- Each word is represented by a d -dimensional vector (common size 50 - 1,000) => more efficient than the low-density, high dimensional vectors of a full PPMI model
- Noise/outliers from low-frequency co-occurrences have less impact (generalizes better than full PPMI)

Predict Models

- Use/inspired by machine learning methods
- Optimize predicting which words occur together

more later....

Applications & Evaluations

Evaluating Semantic Models

- Intrinsic evaluation:

Do they provide good representations of meaning?

- Extrinsic evaluation:

Are they useful for analyzing natural language?

Extrinsic Evaluation

- Which models improve individual tasks most?


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

Methods

- Rule based (i.e. count positive negative words for sentiment classification)
- Machine learning:
 - unsupervised (machine identifies patterns or clusters in data)
 - supervised (machine learns from examples)
 - semi-supervised (combination of supervised & unsupervised)

Methods

- Rule based (i.e. count positive negative words for sentiment classification)
 - Machine learning:
 - unsupervised (machine identifies patterns or clusters in data)
 - **supervised (machine learns from examples)**
 - semi-supervised (combination of supervised & unsupervised)
- 
- MOST COMMONLY USED**

Supervised Machine Learning

- Corpus with desired output annotations:
 - training data, development data, evaluation data
- Features that are informative for the desired output:
 - which information is relevant & available?
=> input for machine learning
- Machine learning software identifying patterns:
 - which output should be provided given the input?

ML example NERC

In 1907 kreeg **Johanna Westerdijk** van **professor F.A.F.C. Went** uit **Utrecht** een collectie levende schimmels. Op basis daarvan ontwikkelde zij **het Centraal Bureau voor Schimmelcultures (CBS)**

In 1907 **Johanna Westerdijk** received a collection living molts from **professor F.A.F.C. Went** from **Utrecht**. She used this to develop **het Centraal Bureau voor Schimmelcultures (CBS)**.

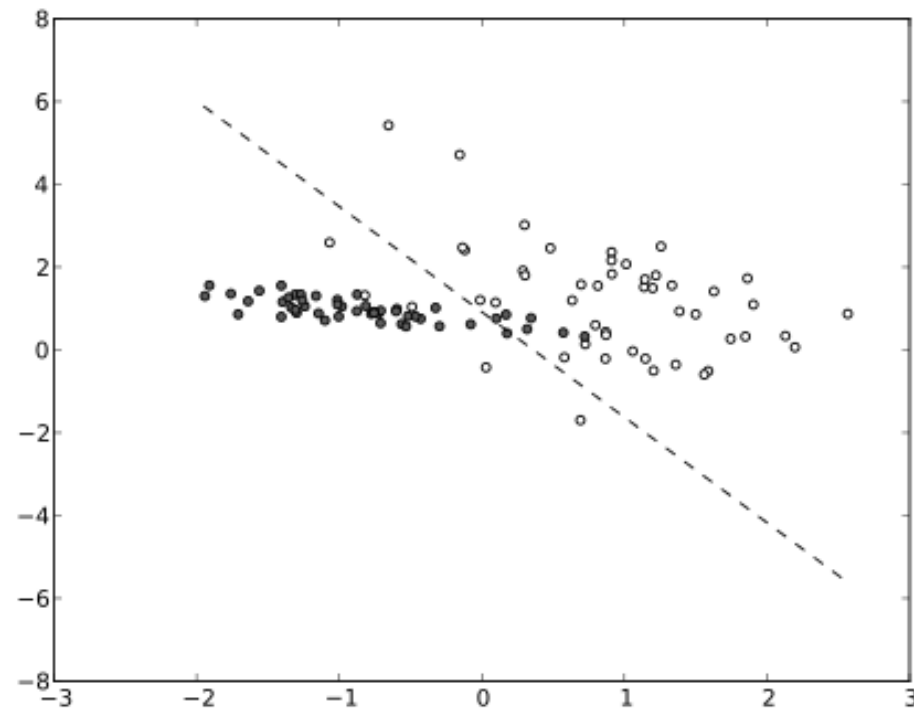
ML example opinion mining

Germans are overwhelmingly opposed to military action in Syria.

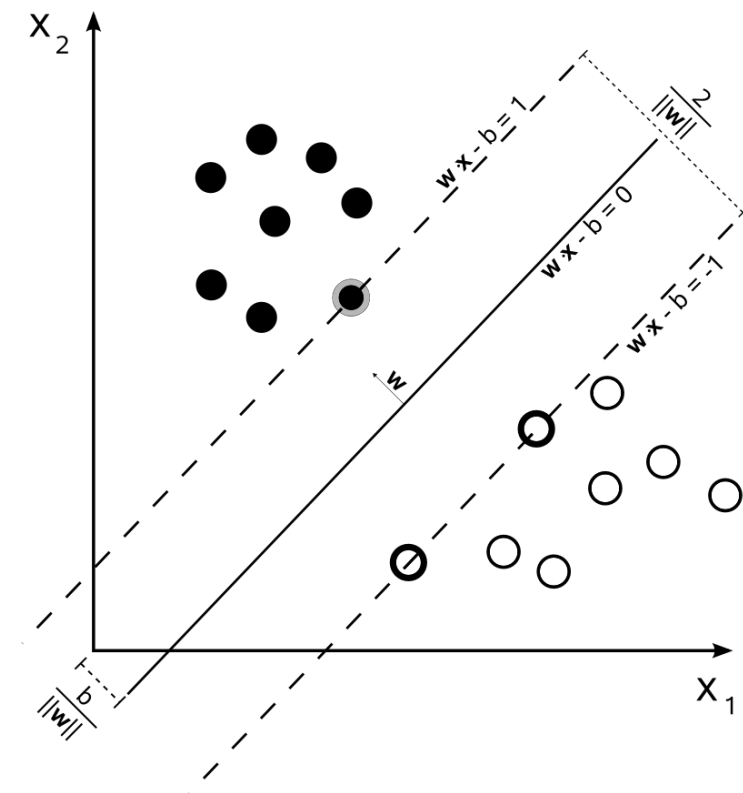
Merkel calls EU stance on Syria “invaluable”.

Obama angry and frustrated about Ebola response

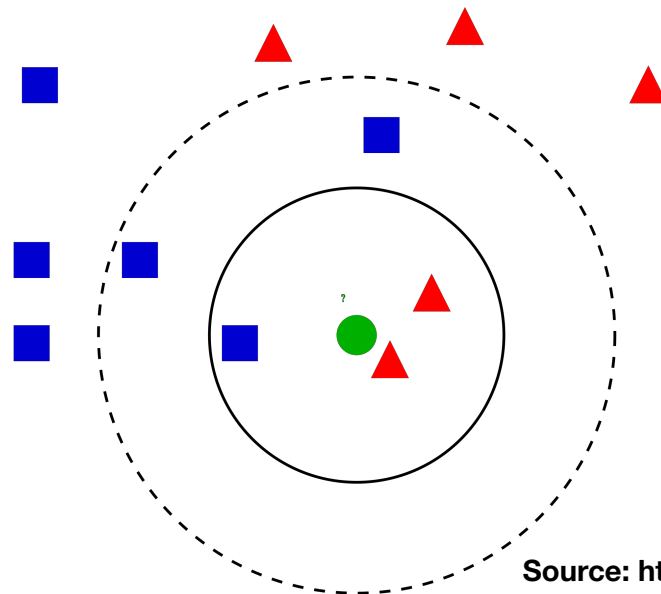
ML methods (examples)



Source: <https://commons.wikimedia.org/wiki/File:Linear-svm-scatterplot.svg>



Source: https://en.m.wikipedia.org/wiki/File:Svm_max_sep_hyperplane_with_margin.png



Source: <https://commons.wikimedia.org/wiki/File:KnnClassification.svg>

ML: basic overview

- Many approaches:
 1. represent features as vectors with numerical values
 2. predict class based on feature vector
- For example:
 - K-nearest neighbor: pick majority class of k-nearest points in space
 - Logistic regression: find hyperplane that separates the data best (minimizing loss)

ML: basic overview

- Generative models: which class was most likely to generate these features

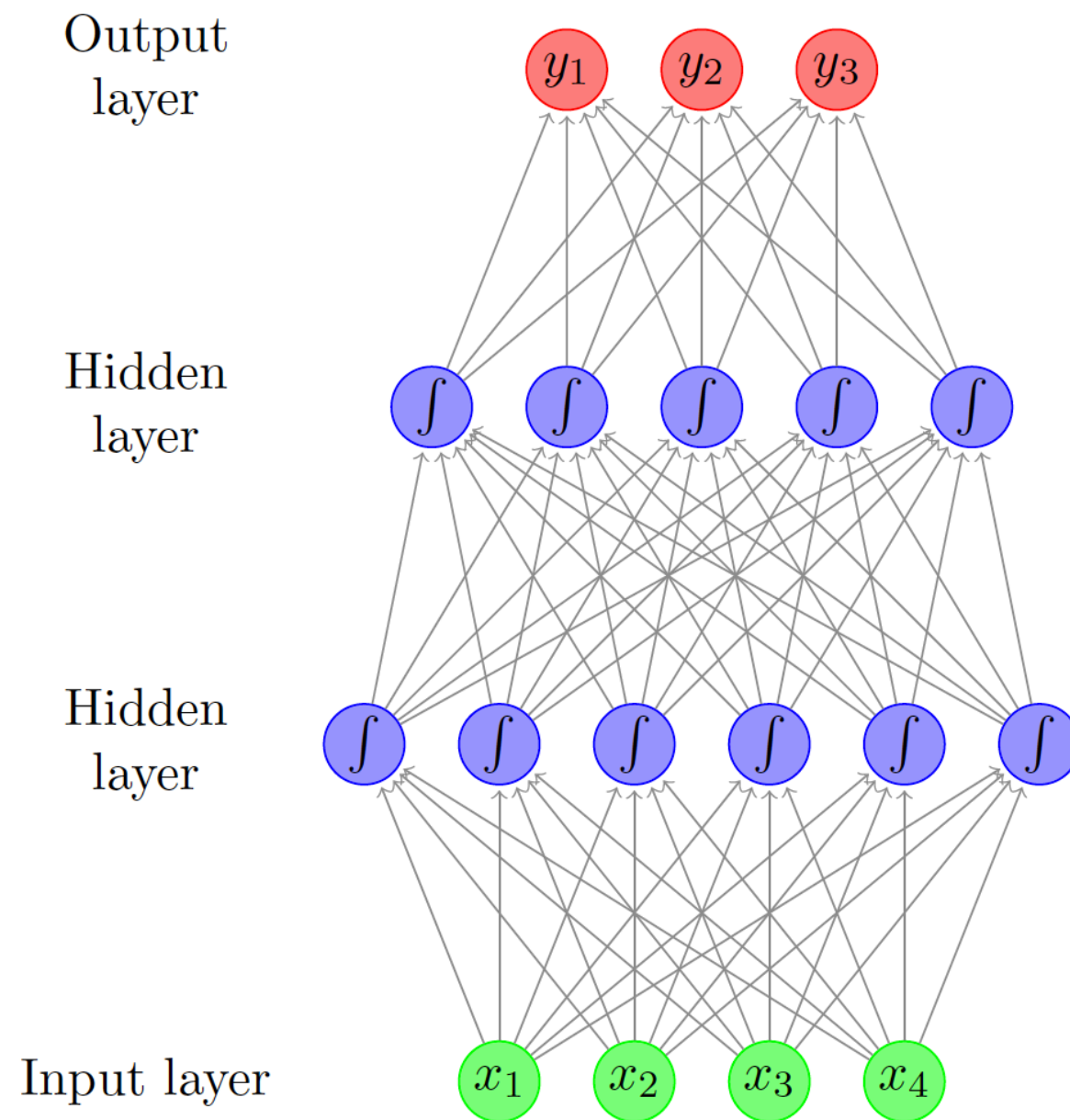
=> does not capture correlation between features
(e.g. capitalized letter + known name counted separately)

- Discriminative model: what is the most likely class given the observed features

=> can capture correlations (capitalized letter or known name will have less weight), but no complex interaction (e.g. subj+passive ~ obj+active)

- Neural networks: can learn complex relations between features

A Basic Neural network



from Goldberg (2015)

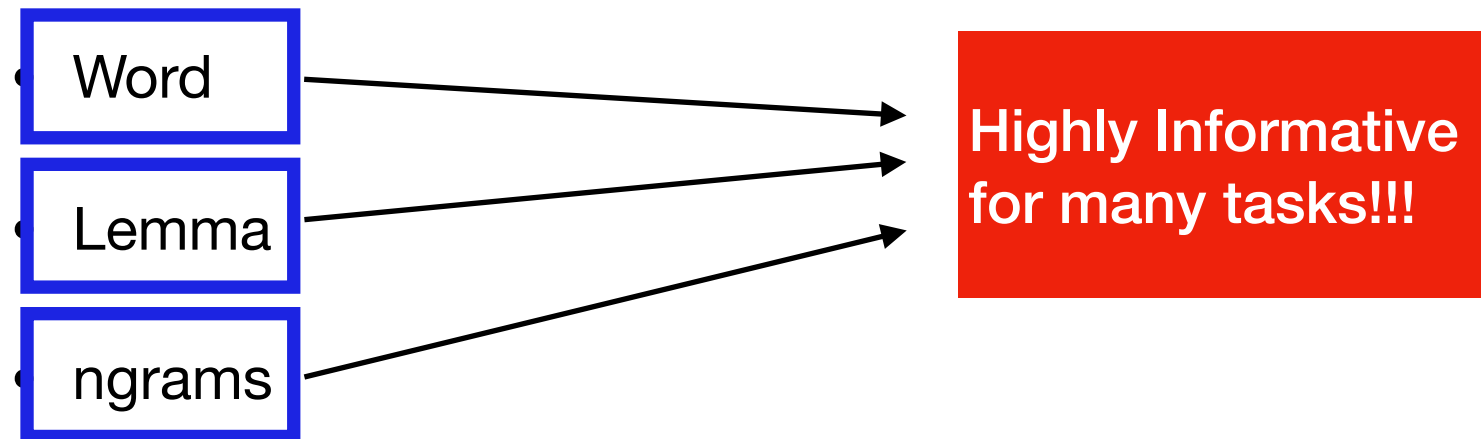
Features

- Common features (for many tasks):
 - POS-tag
 - Word
 - Lemma
 - ngrams
- More advanced:
 - Chunks
 - Syntactic dependencies
 - Word sense

Features

- Common features (for many tasks):

- POS-tag

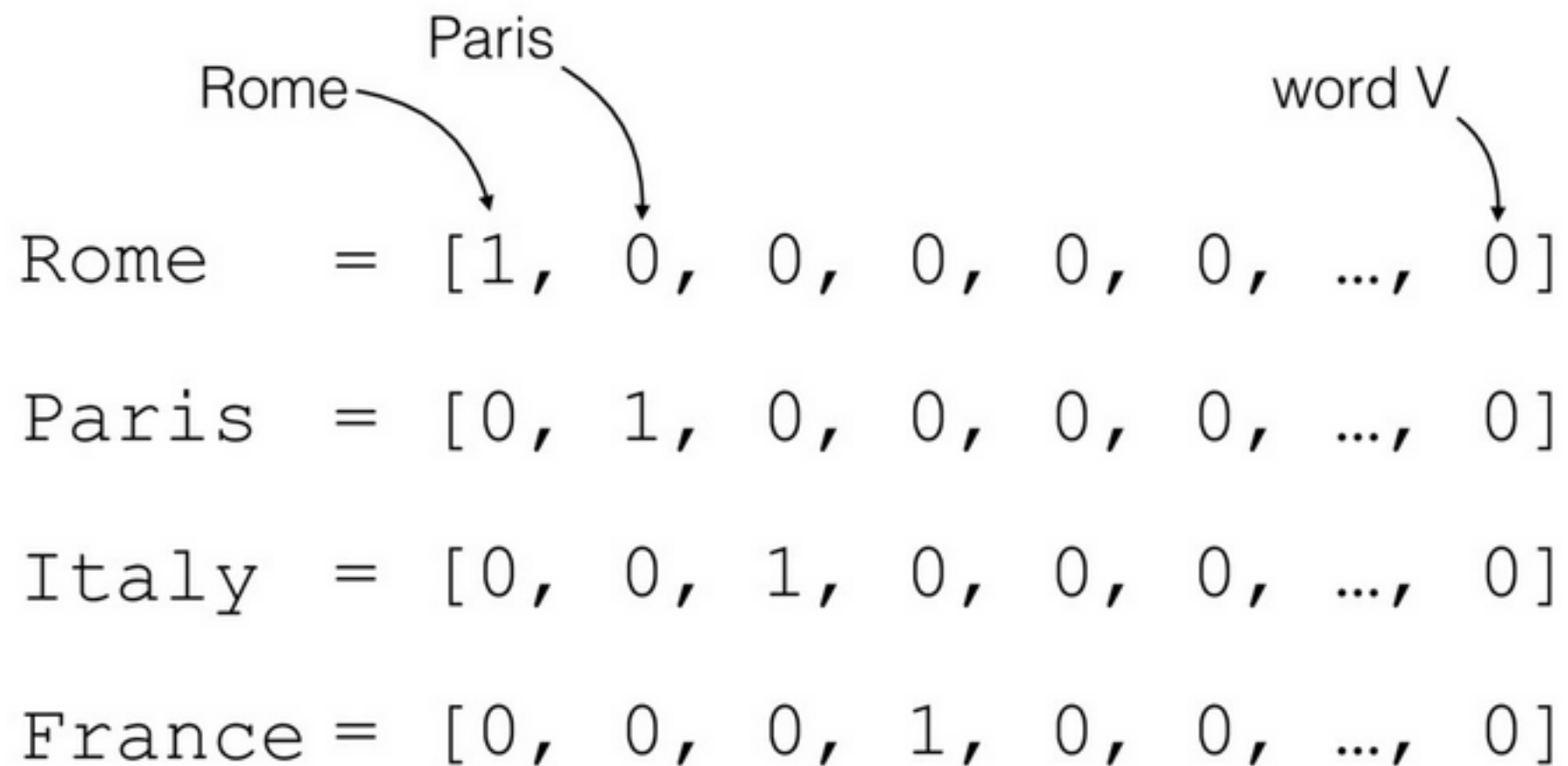


- More advanced:

- Chunks
- Syntactic dependencies
- Word sense

Feature representation

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



One-hot vector

- Consequences of representing words as one-hot vectors:

- very high-dimensional input
- for each feature, each value is (equally) different:

Paris \neq Rome \neq pony \neq smile \neq idea \neq freedom \neq ideas

- Classic solutions:
 - Lemmatizing (turns *ideas* into *idea*)
 - Approximate unknown words by similar words (from WordNet)

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

Predictive Models

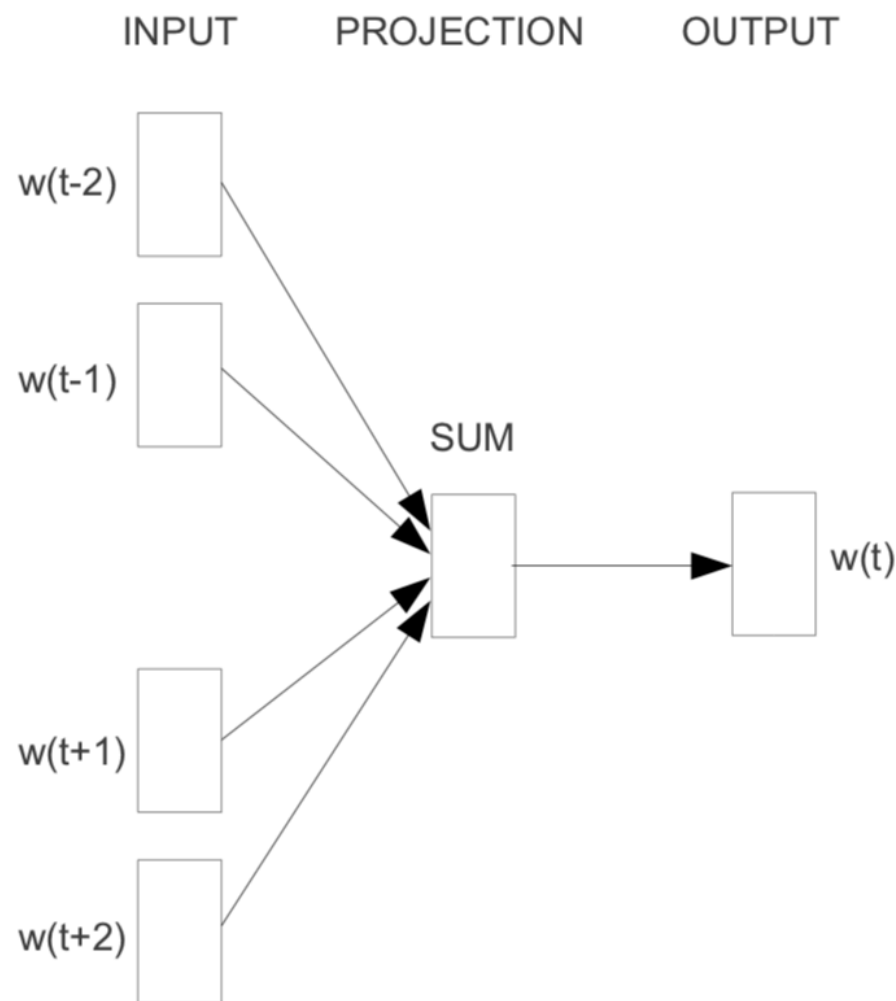
- PPMI: captures desirable properties, but is still sparse and high dimensional
- SVD: generalizes better is high density and lower dimensions, but is inefficient to obtain (and does not perform perfectly)
- Alternative idea: use language modeling as an auxiliary task for creating word embeddings

=> machine learning to **predict** which words occur in each other's context

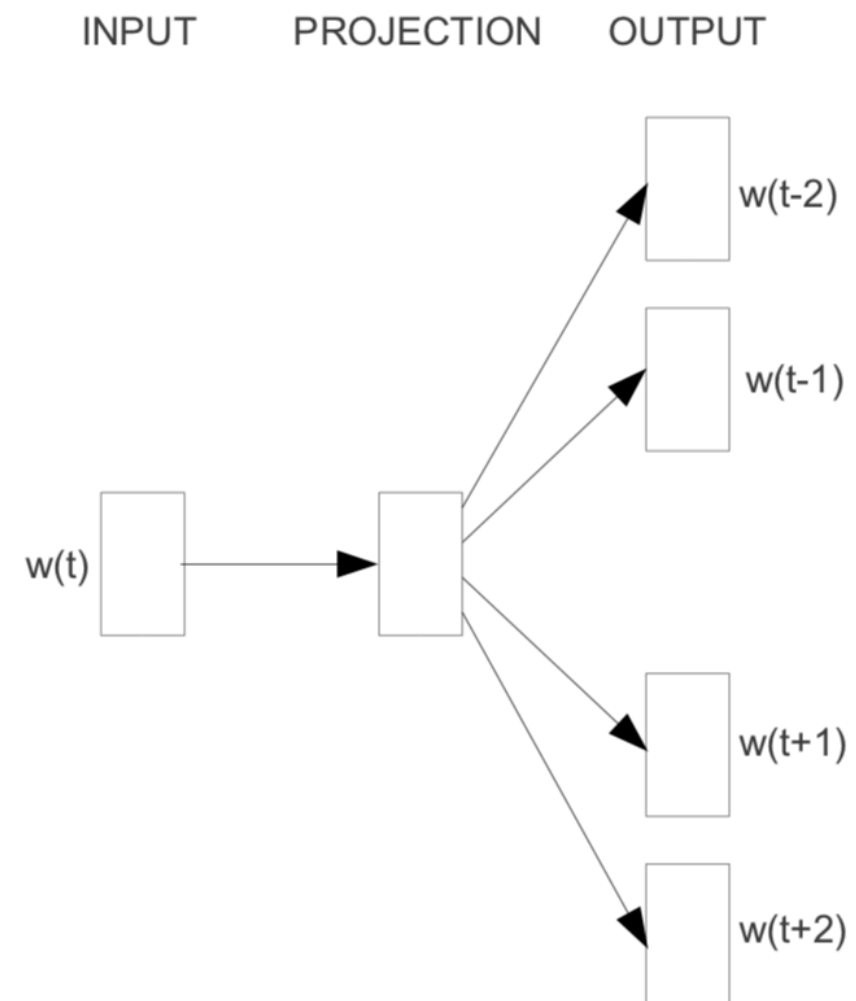
Predictive models

- word2vec (Mikolov et al. 2013a,b):
 - Several methods for creating embeddings
 - All start from randomly initiated vectors with a preset number of dimensions
 - Two vocabulary matrices: one for the words, one for contexts
 - Models: CBOW & Skipgram
 - Training: hierarchical softmax & negative sampling
 - Preprocessing: dynamic context windows, subsampling, delete rare words

CBOW vs Skipgram



CBOW



Skip-gram

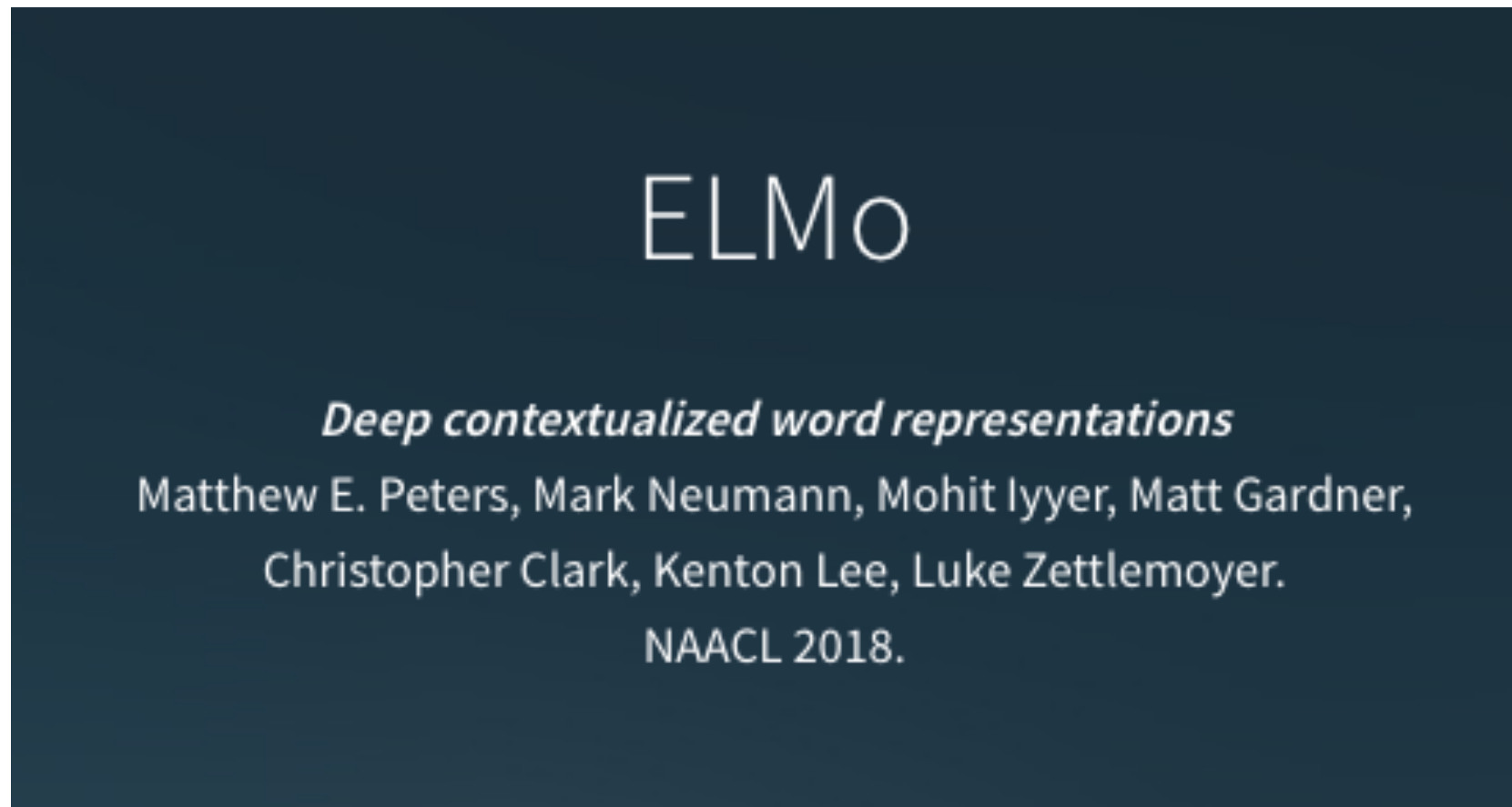
Training

- Hierarchical softmax: efficient way to determine most probable context given a word (or vice versa) over the whole model
- Negative sampling: distinguish the actual context words from k other words (randomly chosen)

Result

- d -dimensional word & context embeddings
- distributional model: the vectors representing words are kept

Latest Developments



<https://allennlp.org/elmo>

- Easy to use (with advanced context selection & neural network for learning)
- State-of-the-art for many NLP tasks

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)

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.
- Hill, F., Reichart, R. and Korhonen, A., 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4), pp.665-695.
- 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.
- Shaffy, Athif (2017) Vector Representation of Text for Machine Learning. <https://medium.com/@athif.shaffy/one-hot-encoding-of-text-b69124bef0a7>