PA164 Natural Language Learning

Lecture 03: Distributional Semantics, LSA and Word Embeddings

Vít Nováček

Faculty of Informatics, Masaryk University

Autumn, 2023

Outline

1. Distributional Semantics
2. Latent Semantic Analysis
3. Word Embeddings
4. Useful References

Historical Notes on Distributional Semantics

- Based on the distributional hypothesis in linguistics
  - Popularised by J. R. Firth in the 1950s
  - *"You shall know a word by the company it keeps."*
- The key assumption, in a more elaborate way:
  - The more semantically similar two words are,
  - the more distributionally similar they will be in turn,
  - and thus will also tend to occur in similar linguistic contexts.
- The distributional hypothesis is the basis for statistical semantics.
- Lately it has been relatively widely studied in other fields, though
  - Cognitive science, language learning, etc.

Illustrative Example of the Distributional Hypothesis

SENTENCE 1: `Colorless green ideas sleep furiously.`

- Context (± 1) of the word `green`: { `Colorless, ideas` }
- Context (± 1) of the word `sleep`: { `ideas, furiously` }

SENTENCE 2: `Colorless red ideas nap furiously.`

- Context (± 1) of the word `red`: { `Colorless, ideas` }
- Context (± 1) of the word `nap`: { `ideas, furiously` }

CONCLUSION:

- `green` is semantically close (identical, actually) to `red`
- `sleep` is semantically close (identical, actually) to `nap`

Distributional vs. Formal Semantics

- Formal semantics studies grammatical meaning using formal tools
  - Building on fields like mathematical logics and theoretical computer science
  - Revolving around central concepts like truth conditions or compositionality
- Distributional semantics is arguably no less formal than the formal one
- Only the key assumptions and formalisms differ
  - Statistics and linear algebra instead of logics
  - Words and phrases instead of structures
  - Similarities instead of truth conditions
- Quite like the classic AI conflict between "neats" and "scruffies"
- Doesn't mean the approaches can not (or should not) be reconciled!

Example Formalisation – A Co-Occurrence Matrix

w = words

c = contexts

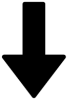
f_{ij} = frequency of cooccurrence

	C1	C2	C3	C4	C5
W1	1	0	0	2	0
W2	0	4	1	0	0
W3	2	0	0	1	0

[1] Toumouh, Adil, Dominic Widdows, and Ahmed Lehireche. "Using Word Space Models for Enriching Multilingual Lexical Resources and Detecting the Relation Between Morphological and Semantic Composition." International Conference on Web and Information Technologies (ICWIT'08). 2008.

Example Formalisation – A Typed Co-Occurrence Tensor

word	*link*	*word*	*weight*	*word*	*link*	*word*	*weight*
marine	own	bomb	40.0	sergeant	use	gun	51.9
marine	use	bomb	82.1	sergeant	own	book	8.0
marine	own	gun	85.3	sergeant	use	book	10.1
marine	use	gun	44.8	teacher	own	bomb	5.2
marine	own	book	3.2	teacher	use	bomb	7.0
marine	use	book	3.3	teacher	own	gun	9.3
sergeant	own	bomb	16.7	teacher	use	gun	4.7
sergeant	use	bomb	69.5	teacher	own	book	48.4
sergeant	own	gun	73.4	teacher	use	book	53.6



	j=1:own	*j=2:use*	*j=1:own*	*j=2:use*	*j=1:own*	*j=2:use*
	k=1:bomb		*k=2:gun*		*k=3:book*	
i=1:marine	40.0	82.1	85.3	44.8	3.2	3.3
i=2:sergeant	16.7	69.5	73.4	51.9	8.0	10.1
i=3:teacher	5.2	7.0	9.3	4.7	48.4	53.6

[2] Baroni, Marco, and Alessandro Lenci. "Distributional memory: A general framework for corpus-based semantics."

Computational Linguistics 36.4 (2010): 673-721.

Outline

1 Distributional Semantics

2 Latent Semantic Analysis

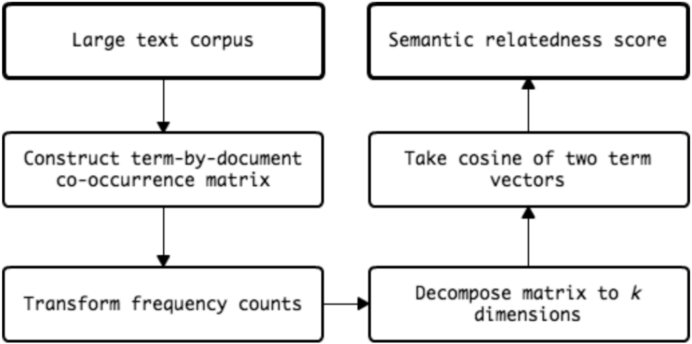
3 Word Embeddings

4 Useful References

Historical Notes on Latent Semantic Analysis

- Arguably the first major success of the "distributional movement"
- Motivated by and applied to the field of information retrieval
  - Given a user query and a corpus of texts,
  - return a text most relevant to the query.
- First described in detail in a 1988 US patent (no. 4,839,853)

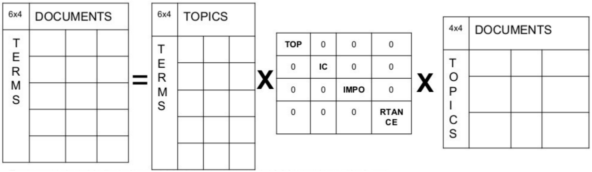
The Gist of LSA



[3] Ryan, James O. "A system for computerized analysis of verbal fluency tests." (2013).

Formalisation of LSA

- It's all about decomposition of a document-term matrix, really
- Specifically, about singular value decomposition (SVD)
  - Given a (big) document-term matrix X,
  - find smaller matrices U, Σ, V such that $X = U\Sigma V^t$.



- Example space savings
  - With 5 topics, 1,000 documents and 1,000 words in a vocabulary,
  - the full document-term matrix size is 10^6 values,
  - but the decomposed matrices correspond to ca. 10^4 $(2 \cdot 5 \cdot 1000 + 5)$

[4] Kovanović, V., and Joksimović, S., and Gašević, D. "Topic Modeling for Learning Analytics Researchers." A LAK15 Tutorial (2015).

Applications of LSA

- Information retrieval by
  - translating query into the low-dimensional space,
  - and finding matching documents
- Comparing documents (using the low-dimensional space)
- Cross-language information retrieval
- Finding relations between terms (synonymy and polysemy)
- Expanding the feature spaces of text mining systems
- Analyzing word associations in a corpus

Notes on LSA Implementation(s)

- The decomposition is an expensive operation
  - Exact methods available (e.g., Lanczos algorithm), but often intractable in practice
  - It's more practical to use incremental, low-memory algorithms (c.f. `gensim`)
  - Neural methods also a viable alternative (for instance Hebbian learning)
- Despite the conceptual simplicity and vast popularity, LSA has limitations:
  - Unclear semantic interpretation of the resulting compressed dimensions
  - Polysemy tends to get "squashed" in the low-dimensional space
  - Bag of words model doesn't capture much of the texts' structure
  - The method expects Gaussian distribution, while in fact Poisson distribution has been observed (addressed by probabilistic LSA)

Outline

1 Distributional Semantics

2 Latent Semantic Analysis

3 Word Embeddings

4 Useful References

History and Gist of Word Embeddings

- Inherently related to distributional semantics
- Outcome of incremental developments in formalising so called "semantic spaces"
  - (Relatively) low-dimensional metric spaces
  - Easier to represent, less noisy and more amenable to computation than the original text
  - The embedding spaces preserve the meaning of the words or phrases
  - Similarities (or distances) in the embedding space reflect the semantic similarity in the original text
- Major historical milestones
  - Vector space model in information retrieval (ca. 1960s)
  - LSA and random indexing in late 1980s
  - In 2000s, Bengio et al. came with first neural approaches
  - In 2013, word2vec by Mikolov et al. kick-started development of modern, highly efficient models

The Two Approaches to Word Embeddings

1. Representation of terms via documents they occur in
   - An extensional representation motivated by information retrieval (c.f. LSA)
   - A token vector is based on a "bag of documents" that contain the token
2. Representation of terms via other terms they occur with
   - An independent approach developed by the computational linguistics community
   - A token vector is based on a "bag of tokens" that co-occur with it in a common linguistic context
3. Most modern approaches use the latter representation

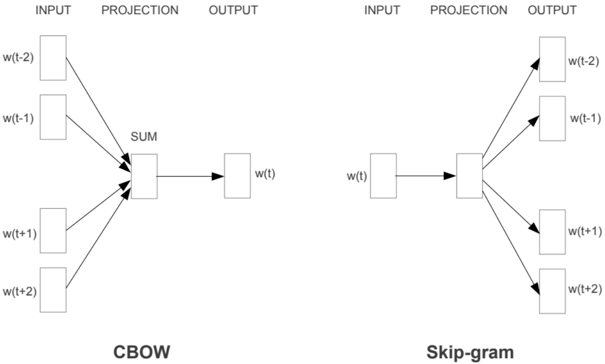
Overview of Modern Word Embedding Approaches

- Specifically focused embedding models
  - Typically based on shallow neural networks and/or optimisation algorithms
  - Trained to produce embeddings directly
  - Examples: word2vec / fastText, GloVe
- More general language models
  - Typically using attention-based deep neural architectures (transformers)
  - Embeddings are a by-product of learning the general language model
  - Examples: ELMo, BERT

word2vec / fastText – the Gist

- Relatively simple log-linear models (2-layer neural networks)
- Words in text are parametrised by vectors associated with them
  - Those are the embeddings
  - Arbitrarily chosen number of elements (typically 100-1,000)
  - No direct relationship to the semantics (initialised, then learned)
- Interactions between word vectors are modelled by a simple function
  - Called a scoring or aggregation function (e.g., scalar product)
- Two dual models
  1. Continuous bag of words - a sliding window in which the context (e.g., 4 previous words, 4 next words) is used to predict the central word (masked in the training stage)
  2. Continuous skip-gram - also uses a sliding window, only the task is to use the central word to predict the context words
- Innovative validation protocols
  - Semantic-syntactic word relatedness benchmarks

word2vec / fastText – the Two Core Models



[5] Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).

word2vec / fastText – Insight into the Training Process

- Specifically, the skip-gram model with negative sampling
- Assuming a corpus of words $w_1, \ldots, w_T$
- The objective is then to maximize the following log-likelihood:
  - $\sum_{t=1}^{T} \sum_{c \in \mathcal{C}_t} \log p(w_c | w_t)$, where $\mathcal{C}_t$ is the context of t
- The probability $p(w_c | w_t)$ can be defined using softmax:
  - $p(w_c | w_t) = \frac{e^{s(w_t, w_c)}}{\sum_{j=1}^{W} e^{s(w_t, j)}}$, where s is the scoring function and W is the size of the vocabulary
- Thus the objective can be rewritten as:
  - $\sum_{t=1}^{T} \left[\sum_{c \in \mathcal{C}_t} \lambda(s(w_t, w_c)) + \sum_{n \in \mathcal{N}_{t,c}} \lambda(-s(w_t, n)) \right]$, where λ is the logistic loss and $\mathcal{N}_{t,c}$ is a set of negative examples sampled from the vocabulary
- This is then optimised using gradient descent

word2vec / fastText – Final Remarks

- The major optimisations and extensions used:
  - Getting rid of the non-linear hidden layer from previous neural models
  - Hierarchical sotfmax via Huffman trees
  - Sub-sampling of relatively frequent words
  - Adding sub-word features
- Validation benchmark examples:
  - v("brother") - v("man") + v("woman") $\sim$ v("sister")
  - v("biggest") - v("big") + v("small") $\sim$ v("smallest")
  - France is to Paris as Germany is to Berlin, mouse is to mice as dollar is to dollars, etc.
- Limitations:
  - Reasons for success poorly understood
  - Largely disregard corpus statistics due to local context windows
  - Very sensitive to hyper-parameters
  - In fact, the same set of hyper-parameters applied to different models can result in very similar performance

GloVe – the Gist

- Motivated by the complementary shortcomings of methods like LSA or word2vec
- The goal:
  - Leverage both corpus statistics and localised distributional features
- The solution:
  - Train on global word-word co-occurrence counts (or rather their ratios)
  - Design a bespoke log-bilinear regression model (i.e., loss function)
  - Cast and optimise the model as a weighted least squares problem

GloVe – Insight into the Training Process

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

- Assume X as the matrix of word-word co-occurrence counts
  - X_{ij} – number of times word j occurs in the context of word i
  - $X_i = \sum_k X_{ik}$ – number of times any word appears nearby word i
  - $P_{ij} = P(j|i) = X_{ij}/X_i$ – probability that word j appears nearby word i
- Most general model: $F(w_i, w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}$
- Refined loss function (already cast as the least squares problem):
  - $J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij})^2$, where V is the size of the vocabulary

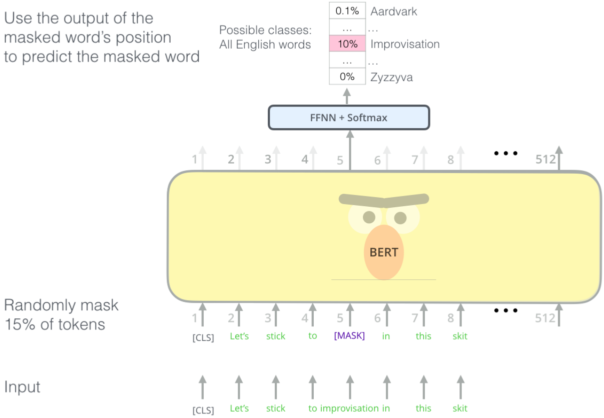
[6] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global vectors for word representation."

Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.

Language Models – the Gist

- In general, a statistical language model is a probability distribution over sequences of words
- Given a sequence of length m, it assigns a probability $P(w_1, \ldots, w_m)$ to the whole sequence
- Thus it can be, for instance,
  - trained on a corpus of natural language tokens to
  - predict the probability of the next token
  - based on a sequence of previous tokens.
- Modern language models are typically
  - trained in an unsupervised manner (using masking of tokens)
  - on very large natural language corpora
  - using transformers (i.e., deep neural architectures with attention mechanism).
- A sort of by-product of the training process are localised word embeddings (as parametrised tokens)

Language Models – Insight into the Training Process



[7] Jay Alammar. "The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)." The `http://jalammar.github.io/` blog (2018-2021).

Outline

1 Distributional Semantics

2 Latent Semantic Analysis

3 Word Embeddings

4 Useful References

Further Readings on Distributional Semantics

- Sahlgren, Magnus. "The distributional hypothesis." Italian Journal of Disability Studies 20 (2008): 33-53.
- Baroni, Marco, and Alessandro Lenci. "Distributional memory: A general framework for corpus-based semantics." Computational Linguistics 36.4 (2010): 673-721.
- Bruni, Elia, et al. "Distributional semantics in technicolor." Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2012.
- Grefenstette, Edward. "Towards a formal distributional semantics: Simulating logical calculi with tensors." arXiv preprint arXiv:1304.5823 (2013).

Further Readings on Latent Semantic Analysis

- Deerwester, Scott, et al. "Indexing by latent semantic analysis." Journal of the American society for information science 41.6 (1990): 391-407.
- Dumais, Susan T. "Latent semantic analysis." Annual review of information science and technology 38.1 (2004): 188-230.
- Hofmann, Thomas. "Probabilistic latent semantic indexing." Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. 1999.
- Hofmann, Thomas. "Unsupervised learning by probabilistic latent semantic analysis." Machine learning 42.1 (2001): 177-196.

Further Readings on Word Embeddings

- The "word2vec papers":
  - Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).
  - Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." Advances in neural information processing systems. 2013.
- The "fastText papers":
  - Joulin, Armand, et al. "Bag of tricks for efficient text classification." arXiv preprint arXiv:1607.01759 (2016).
  - Bojanowski, Piotr, et al. "Enriching word vectors with subword information." Transactions of the Association for Computational Linguistics 5 (2017): 135-146.
- The "GloVe paper":
  - Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global vectors for word representation." Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.

Further Readings on Language Models

- Stolcke, Andreas. "Bayesian learning of probabilistic language models." Diss. University of California, Berkeley, 1994.
- Zhai, ChengXiang. "Statistical language models for information retrieval." Synthesis lectures on human language technologies 1.1 (2008): 1-141.
- Bengio, Yoshua. "Neural net language models." Scholarpedia 3.1 (2008): 3881.
- Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).
- Peters, Matthew E., et al. "Deep contextualized word representations." arXiv preprint arXiv:1802.05365 (2018).