LINFO2263: Vector Semantics and Word Embeddings

Pierre Dupont

Outline

1. Vector semantics from co-occurrence matrices
2. Learning dense word embeddings

Outline

Lexical semantics

Distributional hypothesis

- The meaning of a word **is** its use in the language

  [Wittgenstein, 1953]

- Language use can be characterized by counting how often other words occur in the context of appearance of a specific word

Vector semantics and co-occurrence matrix

- Co-occurrence frequencies between a (target) word and other (context) words can be stored in a vector representing the target word meaning
- The context can be a document, a paragraph, a sentence or, simply, a local context before and after the target word

```
... lemon,  a [tablespoon of apricot jam,      a] pinch ...
               c1          c2    t      c3         c4
```

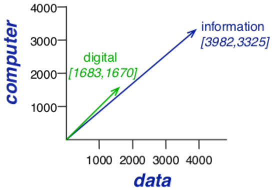
- Such information can be stored in a co-occurrence matrix
  - each row defines the vector associated to a specific target word
  - the number of columns (= the dimensionality of the vector space) depends on the number of contextual words, by default, the whole vocabulary for the task

	aardvark	...	computer	data	result	pie	sugar	...
cherry	0	...	2	8	9	442	25	
strawberry	0	...	0	0	1	60	19	
digital	0	...	1670	1683	85	5	4	
information	0	...	3325	3982	378	5	13	

Illustration from *Speech and Language Processing, Jurafsky and Martin, 3rd ed.*

Vector semantics: example

	aardvark	...	computer	data	result	pie	sugar	...
cherry	0	...	2	8	9	442	25	
strawberry	0	...	0	0	1	60	19	
digital	0	...	1670	1683	85	5	4	
information	0	...	3325	3982	378	5	13	



- 2-D representation here, the real space is $\mathbb{R}^d$ with $d \approx$ vocabulary size
- the similarity between word meanings can be **computed** from vector similarity

Cosine similarity

$$cos(\boldsymbol{v}, \boldsymbol{w}) = \frac{\boldsymbol{v} \cdot \boldsymbol{w}}{\|\boldsymbol{v}\| \|\boldsymbol{w}\|} = \frac{\sum_{i=1}^{d} v_i w_i}{\sqrt{\sum_{i=1}^{d} v_i^2} \sqrt{\sum_{i=1}^{d} w_i^2}}$$

$cos(\boldsymbol{v}, \boldsymbol{w}) = 1$ (is maximal)
$\Leftrightarrow$ the angle between $\boldsymbol{v}$ and $\boldsymbol{w}$ is 0
$\Leftrightarrow$ the relative frequencies of **all** co-occurring words are the same

Illustration from *Speech and Language Processing, Jurafsky and Martin, 3rd ed.*

Weighted and normalized term frequencies

	aardvark	...	computer	data	result	pie	sugar	...
cherry	0	...	2	8	9	442	25	
strawberry	0	...	0	0	1	60	19	
digital	0	...	1670	1683	85	5	4	
information	0	...	3325	3982	378	5	13	

- Actual vector semantics are based on weighted and normalized frequencies because the raw counts are very skewed and not very discriminative
  - words that co-occur frequently (*e.g. pie* nearby *cherry*) look informative
  - yet, very frequent words (*e.g. the, it, they, ...*) co-occur with nearly every other word and are less informative
- Term-Frequency Inverse Document Frequency (TF-IDF) balances both
- TF-IDF comes from information retrieval where the various terms (= words) appear in sets of documents. Here, "documents" refer to the observed local contexts of a target word and the "terms" are the local context words c_j's

```
... lemon,  a [tablespoon of apricot jam,      a] pinch ...
               c1          c2    t      c3        c4
```

TF-IDF weighting applied to Vector Semantics

- $C(w_i, c_j)$ = number of times context word c_j occurs in the local contexts of target word w_i
- **Term frequency**
  The (smoothed) co-occurrence frequency on a log scale

$$tf_{i,j} = \log_{10}\left(C(w_i, c_j) + 1\right)$$

- **Inverse Document Frequency**

$$idf_j = \log_{10}\frac{N}{df_j}$$

  - df_j the number of contextual windows of any target word where this context word c_j occurs
  - N the total number of contextual windows for all target words
  - function words (*the, a, it, ...*) have a high document frequency $\Rightarrow$ a low inverse document frequency
- **TF-IDF** weighted frequency $w_{i,j} = tf_{i,j} \times idf_j$

Positive Pointwise Mutual Information (PPMI)

An alternative to TF-IDF

- **Pointwise Mutual Information** measures how much two words co-occur **more** than expected by chance
  For a word w and a context word c, $PMI(w,c) = \log_2 \frac{P(w,c)}{P(w)P(c)}$
- A negative PMI would mean w and c co-occur less than expected by chance but getting reliable estimates of this is difficult (huge corpora are required to estimate rare events reliably)
- **Positive PMI** replaces all negative PMI values by 0

$$PPMI(w,c) = \max\left(\log_2 \frac{P(w,c)}{P(w)P(c)}, 0\right)$$

PPMI estimation

- $C(w_i, c_j)$ = number of times context word c_j occurs in the local contexts of target word w_i
- vocabulary V = the set of words
- set of context words C, possibly $C = V$
- additive smoothing hyper-parameter: $\varepsilon \approx \frac{1}{|V|}$ (e.g. $\varepsilon = 10^{-4}$)

$$\hat{P}(w_i, c_j) = \frac{C(w_i, c_j) + \varepsilon}{\sum_{i=1}^{V} \sum_{j=1}^{C} \left[C(w_i, c_j) + \varepsilon\right]}$$

$$\hat{P}(w_i) = \sum_{j=1}^{C} \hat{P}(w_i, c_j) \qquad \hat{P}(c_j) = \sum_{i=1}^{V} \hat{P}(w_i, c_j)$$

$$PPMI(w_i, c_j) = \max\left(\log_2 \frac{\hat{P}(w_i, c_j)}{\hat{P}(w_i)\hat{P}(c_j)}, 0\right)$$

Sparse word embeddings

- dimensions of co-occurrence based word embeddings (from TF-IDF or PPMI) have a direct interpretation = the identities of context words appearing in a local context of target words
- TD-IDF or PPMI define sparse word embeddings
  - the word meanings are embedded in a very high dimensional space with lots of zeros
  - a term frequency is 0 if a context word never occurs in the local context of a target word
  - a negative pointwise mutual information has been replaced by 0 (or the co-occurrence probability is exactly the one expected by chance)

	computer	data	result	pie	sugar
cherry	0	0	0	4.38	3.30
strawberry	0	0	0	4.10	5.51
digital	0.18	0.01	0	0	0
information	0.02	0.09	0.28	0	0

A PPMI matrix example

Illustration from *Speech and Language Processing, Jurafsky and Martin, 3rd ed.*

Outline

Learning dense word embeddings

- Use vector space of smaller dimensionality ($50 \ldots 300$) instead of the vocabulary size ($\approx 50,000$)
  - fewer parameters to represent word meanings may be enough
- **Learn** dense word embeddings rather than defining them from (weighted) co-occurrence counts
  - dimensions of the vector space no longer represent co-occurring words but rather abstract dimensions of meaning
  - dimensions *could* represent notions such as *positive/negative sentiment*, *trendy/old-fashioned concept*, ...
  - in practice, these abstract dimensions are automatically defined by a learning algorithm
- Word embeddings are learned to solve a specific NLP task: sentiment analysis, sentence completion (Shannon's game), ...and sometimes reused for another task: translate, ...
- Learned word embeddings are typically dense (mostly non-zeros)

Word2Vec

[Mikolov, et al., 2013]

1. Learn skip-grams embedding to solve a binary prediction task: is word *w* likely to occur in a context of the target word *t*?

```
... lemon,  a [tablespoon of apricot jam,     a] pinch ...
               c1         c2     t     c3       c4
```

   - Treat target words and neighboring context words (skip bi-grams) as positive examples
   - Randomly sample other words from the vocabulary to form negative examples

positive examples +

t	c
apricot	tablespoon
apricot	of
apricot	jam
apricot	a

negative examples -

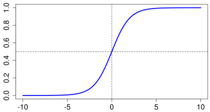
t	c	t	c
apricot	aardvark	apricot	seven
apricot	my	apricot	forever
apricot	where	apricot	dear
apricot	coaxial	apricot	if

2. Train a logistic regression as a classifier to discriminate $+/-$ examples
3. Use regression weights as word embeddings

Illustration from *Speech and Language Processing, Jurafsky and Martin, 3rd ed.*

Word2Vec: logistic regression

- Given a target word t (*e.g.* `apricot`) and a candidate context word c (*e.g.* `jam` or `aardvark`), define the probability $P(+|t,c)$ that c is a positive example (c is a real context word for t)
- $P(+|\boldsymbol{t},\boldsymbol{c})$ is a function of the vectors $\boldsymbol{t}$ and $\boldsymbol{c}$ representing t and c
  - $\boldsymbol{t} \cdot \boldsymbol{c}$ measures the similarity between t and c (an unscaled cosine)
  - this measure is squashed between $[0,1]$ by a sigmoid function
    $P(+|\boldsymbol{t},\boldsymbol{c}) = \sigma(\boldsymbol{t} \cdot \boldsymbol{c})$ with $\sigma(x) = \frac{1}{1+\exp^{-x}} = \frac{\exp^x}{\exp^x + 1}$



Sigmoid a.k.a **Logistic** function

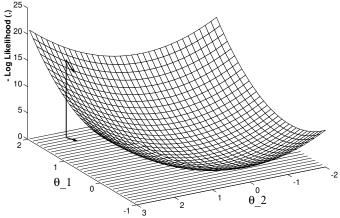
- $P(-|\boldsymbol{t},\boldsymbol{c}) = 1 - P(+|\boldsymbol{t},\boldsymbol{c})$
- Start from **randomly chosen vectors** in d dimensions (an hyper-parameter fixed by the designer)
- Greedily **optimize them** to better fit the training data

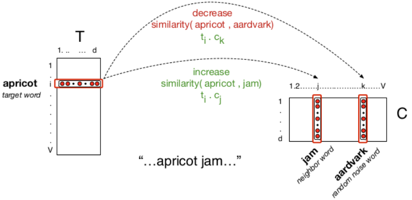
Learning Word2Vec embeddings

- **Model parameters** θ: vectors $\boldsymbol{t}$ and $\boldsymbol{c}$ for all target/context words
- **Log-likelihood**:

$$LL(\theta) = \sum_{t,c \in +} \log P(+|\boldsymbol{t}, \boldsymbol{c}) + \sum_{t,c \in -} \log P(-|\boldsymbol{t}, \boldsymbol{c})$$

  - measures the fit to the positive and negative training examples
  - look for the parameters maximizing $LL(\theta)$ or, equivalently, minimizing $-LL(\theta)$ called a **loss**
  - standard optimization relies on gradient descent





- **Final result**: d-dimensional embeddings for T and C
  - keep just T, or add them $t_i + c_i$, or concatenate them (a $2d$ solution)

Semantic properties of word embeddings

Popular word embeddings, such as **Word2Vec** [Mikolov, et al., 2013] or **GloVe** [Pennington et al., 2014], are effective ways to represent word meanings by vectors

- vector arithmetic allows to combine meanings!!

  $\overrightarrow{king} - \overrightarrow{man} + \overrightarrow{woman} \approx \overrightarrow{queen}$

  $\overrightarrow{Paris} - \overrightarrow{France} + \overrightarrow{Italy} \approx \overrightarrow{Rome}$

- but word embeddings include gender or racist biases, most probably present in the corpora they are trained from

  $\overrightarrow{computer\_scientist} - \overrightarrow{man} + \overrightarrow{woman} \approx \overrightarrow{homemaker}$

  (*ménagère, femme au foyer*...)

Summary

- The meaning of a word is its use in a language $\Rightarrow$ co-occurring words
- Word meanings can be represented by vectors
  - constructed from weighted co-occurrence counts
    $\Rightarrow$ sparse word embeddings (TF-IDF, Pointwise Mutual Information)
  - automatically learned to get abstract dimensions
    $\Rightarrow$ dense word embeddings (Word2Vec, GloVe, . . .)

Further Reading

Jurafsky D. and Martin J.H.
Speech and Language Processing, 3rd edition (draft)
chapters (5 and) 6.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J.
Distributed representations of words and phrases and their compositionality
Advances in Neural Information Processing Systems (NIPS), pp. 3111—3119, 2013.

Pennington, J., Socher, R., and Manning, C. D.
GloVe: Global Vectors for Word Representation
Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543, 2014.

Wittgenstein, L.
Philosophical Investigations. (Translated by Anscombe, G.E.M.).
Blackwell, 1953