Vector Space Models of Lexical Meaning
Seminar Word Embeddings for NLP and IR

Polina Stadnikova

Saarland University

9th November 2017

Outline

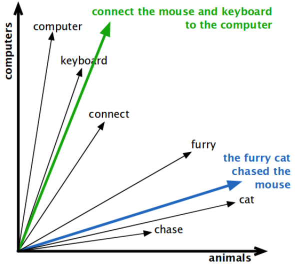
How do we represent word meanings formally?

Set theory based vs. Distributional

Set theory

All cats chase mice:
$\forall x \forall y((cat'(x) \wedge mouse'(y)) \rightarrow chase'(x, y))$
Each mouse is connected to a computer:
$\forall x(mouse'(x) \rightarrow \exists y(comp'(y) \wedge connect'(x, y))$

Vector models



Set theory based vs. Distributional

Set theory

Represent object, their properties and relations between them, meanings = constraints

Vector models

Capture distance and similarity, meanings = vectors in a semantic space, **more fine-grained representation**

The basic idea of vector space models

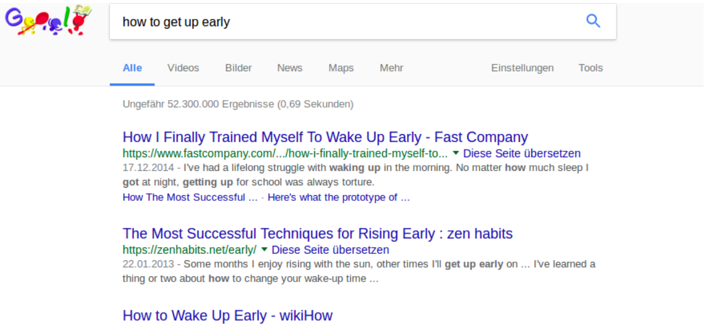
"You shall know a word by the company it keeps!"

— J. R. Firth (1957)

Distributional hypothesis

- Words that occur in similar contexts tend to have similar meanings
- Set of contexts for a word = distribution
- Distribution of the contexts represents the meaning

Document retrieval



- Input: a query
- Output: a set of documents ranked according to their relevance

Ignores the word order and the syntactic dependencies, but works well

Document retrieval

Task = find an overlap between the query and the documents

Vector space approach

- Words = basis vectors
- Queries and documents = vectors in the space

How to measure the similiraty between document and query?

$$Sim(\vec{q}, \vec{d}) = \sum_i q_i * d_i$$

Document retrieval: example

Term vocabulary: ⟨*England, Australia, Pietersen, Hoggard, run, wicket, catch, century, collapse*⟩

Document $d1$: *Australia collapsed as Hoggard took 6 wickets . Flintoff praised Hoggard for his excellent line and length .*

Document $d2$: *Flintoff took the wicket of Australia 's Ponting , to give him 2 wickets for the innings and 5 wickets for the match .*

Query q: { *Hoggard, Australia, wickets* }

$$\overrightarrow{q1} \,.\, \overrightarrow{d1} = \langle 0,1,0,1,0,1,0,0,0\rangle \,.\, \langle 0,1,0,2,0,1,0,0,1\rangle = 4$$

$$\overrightarrow{q1} \,.\, \overrightarrow{d2} = \langle 0,1,0,1,0,1,0,0,0\rangle \,.\, \langle 0,1,0,0,0,3,0,0,0\rangle = 4$$

vector coefficients = term-frequencies

Document retrieval: example

Term vocabulary: ⟨*England, Australia, Pietersen, Hoggard, run, wicket, catch, century, collapse*⟩

Document $d1$: *Australia collapsed as Hoggard took 6 wickets . Flintoff praised Hoggard for his excellent line and length .*

Document $d2$: *Flintoff took the wicket of Australia 's Ponting , to give him 2 wickets for the innings and 5 wickets for the match .*

Query q: { *Hoggard, Australia, wickets* }

$$\vec{q1} \, . \, \vec{d1} = \langle 0,1,0,1,0,1,0,0,0 \rangle \, . \, \langle 0,1,0,2,0,1,0,0,1 \rangle = 4$$

$$\vec{q1} \, . \, \vec{d2} = \langle 0,1,0,1,0,1,0,0,0 \rangle \, . \, \langle 0,1,0,0,0,3,0,0,0 \rangle = 4$$

vector coefficients = term-frequencies

Document retrieval: more sophisticated approach (TF IDF)

Some words have more discriminating power!

The basic idea:

- If a word occurs only in few documents, it provides a stronger evidence for a particular meaning
- $\rightarrow$ should have a higher weight

Document retrieval: more sophisticated approach (TF IDF)

How to weight the words?

- count in how many documents a word occurs (document-frequency)
- coefficients in $=$ term-frequency/document-frequency
- or: coefficients in $=$ term-frequency **inverse document-frequency*

Document retrieval: more sophisticated approach (TF IDF)

How to weight the words?

- count in how many documents a word occurs (document-frequency)
- coefficients in $=$ term-frequency/document-frequency
- or: coefficients in $=$ term-frequency **inverse document-frequency*

$$\overrightarrow{q1} \,.\, \overrightarrow{d1} = \langle 0,1,0,1,0,1,0,0,0 \rangle \,.\, \langle 0,1/10,0,2/5,0,1/100,0,0,1/3 \rangle = 0.41$$

$$\overrightarrow{q1} \,.\, \overrightarrow{d2} = \langle 0,1,0,1,0,1,0,0,0 \rangle \,.\, \langle 0,1/10,0,0/5,0,3/100,0,0,0/3 \rangle = 0.13$$

Document retrieval: more sophisticated approach (TF IDF)

Normalization by length

$$Sim(\vec{q}, \vec{d}) = \cos(\vec{q}, \vec{d})$$

Document retrieval: more sophisticated approach (TF IDF)

Normalization by length

$$Sim(\vec{q}, \vec{d}) = \cos(\vec{q}, \vec{d})$$

Why normalization?

To avoid the bias towards longer documents

Document retrieval: term-document matrix

$$
\begin{array}{l}
 \\
England \\
Australia \\
Pietersen \\
Hoggard \\
run \\
wicket \\
catch \\
century \\
collapse
\end{array}
\begin{array}{c}
\begin{array}{cc} d1 & d2 \end{array} \\
\begin{pmatrix}
0 & 0 \\
1/10 & 1/10 \\
0 & 0 \\
2/5 & 0/5 \\
0 & 0 \\
1/100 & 3/100 \\
0 & 0 \\
0 & 0 \\
1/3 & 0/3
\end{pmatrix}
\end{array}
$$

Why do we need this?

Better understand co-occurrence of the data to define word vectors

Document retrieval: what else is important?

How to reduce the size of a matrix?

Singular Value Decomposition(SVD)

SVD in a nutshell

- factors the original matrix into 3 matrices
- uses the 3 matrices to create a low-rank approximation

Document retrieval: what else is important?

SVD: getting more concrete

- clusters words along a few hundred semantic dimensions
- obtains semantic dimensions from the co-occurrence data
- filters out the noise
- higher-order co-occurrence: similar words appear in similar context

Quick wrap-up

What did we learn so far?

- How to represent a document/a query/a sentence as a vector
- How to measure the similarity between vectors
- How to weight different contextual terms
- How to capture the co-occurrence
- How to reduce large matrices

Context

A narrow definition of similarity

Words are similar if they occur in the same context

~~Context = a set of documents~~

Context = one document, one (partial) sentence

Context

A narrow definition of similarity

Words are similar if they occur in the same context

~~Context = a set of documents~~

Context = one document, one (partial) sentence

A narrow definition of context

Problem: similar words usually do not appear together within one context

$\rightarrow$ shorten context to one word

Context: window method

Topical similarity

$$
\begin{array}{l}
 \\
automobile \\
car \\
soccer \\
football
\end{array}
\begin{array}{c}
\begin{array}{ccccccc} wheel & transport & passenger & tournament & London & goal & match \end{array} \\
\begin{pmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 2 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 2 & 1
\end{pmatrix}
\end{array}
$$

Context: window method

Topical similarity

$$
\begin{array}{l}
\\
automobile \\
car \\
soccer \\
football
\end{array}
\begin{array}{c}
\begin{array}{ccccccc}
wheel & transport & passenger & tournament & London & goal & match
\end{array} \\
\begin{pmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 2 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 2 & 1
\end{pmatrix}
\end{array}
$$

$automobile\ .\ car = 4$

$automobile\ .\ soccer = 0$

$automobile\ .\ football = 1$

$car\ .\ soccer = 1$

$car\ .\ football = 2$

$soccer\ .\ football = 5$

Context: window method

What about synonyms?

- use only few words from both sides of the target as context
- use context words as basis vector
- vector coefficients = weighted co-occurrence frequencies of each context word within the window
- also consider the direction (on which side from the target word?)

Context: pre-processing

Part-of-Speech Tagging

Tag the context, consider syntactic relations $\rightarrow$ vectors contain separate counts for each relation

Lemmatization

lemmatize context words

Context: a fine-grained representation

target word - *goal*

PoS Tagging

Giggs|NNP *scored*|VBD *the*|DT *first*|JJ *goal*|NN *of*|IN *the*|DT *football*|NN *tournament*|NN *at*|IN *Wembley*|NNP ,|, *North*|NNP *London*|NNP .|.

Dependency relations

(ncmod _ *goal first*)
(det *goal the*)
(ncmod _ *tournament football*)
(det *tournament the*)
(ncmod _ *London North*)
(dobj *at Wembley*)
(ncmod _ *scored at*)
(dobj *of tournament*)
(ncmod _ *goal of*)
(dobj *scored goal*)
(ncsubj *scored Giggs* _)

Context: a fine-grained representation

target word - *goal*

Different contexts

Contextual elements for target word *goal* using a 7-word window method:
{*scored, the, first, of, football*}

Contextual elements with parts-of-speech:
{*scored*|VBD, *the*|DET, *first*|JJ, *of*|IN, *football*|NN}

Contextual elements with direction (L for left, R for right):
{*scored*|L, *the*|L, *first*|L, *of*|R, *the*|R, *football*|R}

Contextual elements with position (e.g. 1L is 1 word to the left):
{*scored*|3L, *the*|2L, *first*|1L, *of*|1R, *the*|2R, *football*|3R}

Contextual elements as grammatical relations:
{*first*|`ncmod`, *the*|`det`, *scored*|`dobj`}

Context: more fine-grained?...

Extend syntactic relations

- Create dependency paths
- Basis vectors = whole sequencies of syntactic relation between target and context

Context: more fine-grained?...

Extend syntactic relations

- Create dependency paths
- Basis vectors = whole sequencies of syntactic relation between target and context

Example

$\{\textit{first}|\textit{ncmod}, \textit{the}|\textit{det}, \textit{scored}|\textit{obj}, \textbf{Giggs}|\textbf{subj}\}$

Context: more fine-grained?...

Extend syntactic relations

- Create dependency paths
- Basis vectors = whole sequencies of syntactic relation between target and context

Example

{*first*|*ncmod*, *the*|*det*, *scored*|*obj*, **Giggs**|**subj**}

Problem

Data sparsity: too detailed representation → too small counts

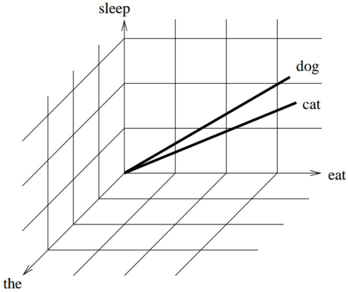
Weighting

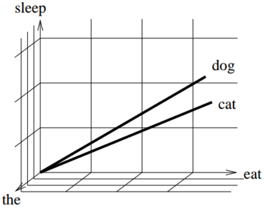
Not all basis vector are equally useful!

A simple approach

- Word frequency / number of document in which word occurs (remember *inverse document frequency (IDF)*?)
- Decrease the weight of highly frequent words

Weighting:IDF





Weighting

An issue with IDF

the same effect to all target words

Weighting

An issue with IDF

the same effect to all target words

Why problematic?

we want to weight different words differently:

wear → jacket

~~wear → car~~

gasoline → car

wear → jacket

Weighting: collocations

The basic idea

Use collocation[a] statistics to discriminate between more and less useful context words

[a] a collocation is a sequence of words which co-occur more often than it would be expected by chance

- make use of parameters (e.g., collocation window)
- set the parameters empirically

Similarity

$$Sim(\vec{q}, \vec{d}) = \frac{\vec{q}.\vec{d}}{||\vec{q}|| \;\; ||\vec{d}||}$$
$$= \cos(\vec{q}, \vec{d})$$

Similarity

$$Sim(\vec{q}, \vec{d}) = \frac{\vec{q}.\vec{d}}{||\vec{q}|| \ \ ||\vec{d}||}$$
$$= \cos(\vec{q}, \vec{d})$$

looks familiar?

Similarity

$$Sim(\vec{q}, \vec{d}) = \frac{\vec{q}.\vec{d}}{||\vec{q}|| \ \ ||\vec{d}||}$$
$$= \cos(\vec{q}, \vec{d})$$

looks familiar?

do not forget the importance of normalization!

Corpus

Which corpus to use?

- depends on the application: web data, query logs, research paper, dictionaries, etc.
- popular: British National Corpus (BNC) $\approx$ 100 million words
- general rule: the more data, the better quality

Corpus

Which corpus to use?

- depends on the application: web data, query logs, research paper, dictionaries, etc.
- popular: British National Corpus (BNC) $\approx$ 100 million words
- general rule: the more data, the better quality

Efficient processing of very large corpora is a big issue in academia

An example output

Ranked synonyms

introduction: launch, implementation, advent, addition, adoption, arrival, absence, inclusion, creation, departure, availability, elimination, emergence, use, acceptance, abolition, array, passage, completion, announcement, ...

evaluation: assessment, examination, appraisal, review, audit, analysis, consultation, monitoring, testing, verification, counselling, screening, audits, consideration, inquiry, inspection, measurement, supervision, certification, checkup, ...

context: perspective, significance, framework, implication, regard, aspect, dimension, interpretation, meaning, nature, importance, consideration, focus, beginning, scope, continuation, relevance, emphasis, backdrop, subject, ...

similarity: resemblance, parallel, contrast, flaw, discrepancy, difference, affinity, aspect, correlation, variation, contradiction, distinction, divergence, commonality, disparity, characteristic, shortcoming, significance, clue, hall-mark, ...

An example output

What can we say about automatically extracted data?

- Not always correlates with the competence of humans
  advent for *introduction*
- Errors
  elimination $\neq$ *introduction*
- Decisions that are hard to explain
  array for *introduction*

Evaliuation

Intrinsic

- Compare against a manually created gold standard
- Disadvantage: some results can be marked as incorrect because of lack of coverage/errors
- Standard measure: precision (accuracy)
  How many synonyms for target words were found and ranked correctly?
- Example: 68% at rank 1 means that 68% top-ranked synonyms from the output were created and top-ranked in the gold standard

Evaliuation

Extrinsic

- Apply the output to a practical task
- Psycholinguistic experiments, crowdsourcing
- An example: semantic priming
- Higher similarity between word and related prime
- Correlation between similarity and reading time

Final wrap-up

Now we know how to...

- **Represent word meanings in vector space**
  Context = basis vectors which define space
  Words to describe = vectors in space

Final wrap-up

Now we know how to...

- **Represent word meanings in vector space**
  Context = basis vectors which define space
  Words to describe = vectors in space

- **Define context of a word**
  Sentence, few words, one word

Final wrap-up

Now we know how to...

- **Represent word meanings in vector space**
  Context = basis vectors which define space
  Words to describe = vectors in space
- **Define context of a word**
  Sentence, few words, one word
- **Weight context words**
  IDF, collocation statistics

Final wrap-up

Now we know how to...

Measure similarity of vectors

...and not to forget the importance of normalization

Final wrap-up

Now we know how to...

Measure similarity of vectors

...and not to forget the importance of normalization

Conduct experiments

Define corpus, application, etc.

Special features of the data obtained from corpora automatically

Final wrap-up

Now we know how to...

Measure similarity of vectors

...and not to forget the importance of normalization

Conduct experiments

Define corpus, application, etc.

Special features of the data obtained from corpora automatically

Evaluate experiments' results

Intrinsic vs. extrinsic, precision

Final wrap-up

We also learned that...

Vector models' design depends on the application

e.g., lexical realtions can help in a search engine

Final wrap-up

We also learned that...

Vector models' design depends on the application

e.g., lexical realtions can help in a search engine

Defining context might be complicated

...and also depends on the application

Final wrap-up

We also learned that...

Vector models' design depends on the application
e.g., lexical realtions can help in a search engine

Defining context might be complicated
...and also depends on the application

We can use differently fine-grained methods to define context
Window method (directed and undirected)
Syntactic relations (PoS tags, dependency relations)

Final wrap-up

Moreover...

Vector representation of word meanings help to extact lexical realtions

synonymy, antonymy, hyponymy, etc.

Final wrap-up

Moreover...

Vector representation of word meanings help to extact lexical realtions

synonymy, antonymy, hyponymy, etc.

We can go into details

...see references

Final wrap-up

Finally...

We learnt more about distributional models in computational semantics!

References

- Clark, Stephen (2015), Vector Space Models of Lexical Meaning. Handbook of Contemporary Semantic Theory — second edition, edited by Shalom Lappin and Chris Fox. Chapter 16, pp.493-522. Wiley-Blackwell.
- Manning, Christopher Hinrich Schutze (1999), Foundations of Statistical Natural Language Processing, The MIT Press, Cambridge, Massachusetts.
- Curran, James R. (2004), From Distributional to Semantic Similarity, Ph.D. thesis, University of Edinburgh.

Thanks for your attention!

Questions?