Word Embeddings

Lexical Resources - Lectures 6 & 7

November 28th, 2018

Contents of the lecture

1. General overview
2. Technical definition
3. What to do with word vectors

Note :

- the teaching materials are available on GitHub: https://github.com/TimotheeMickus/lexical_resources
- You will need to install numpy: pip install numpy
- All exercises are due for next Tuesday, noon, at latest.

Overview

Overview

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

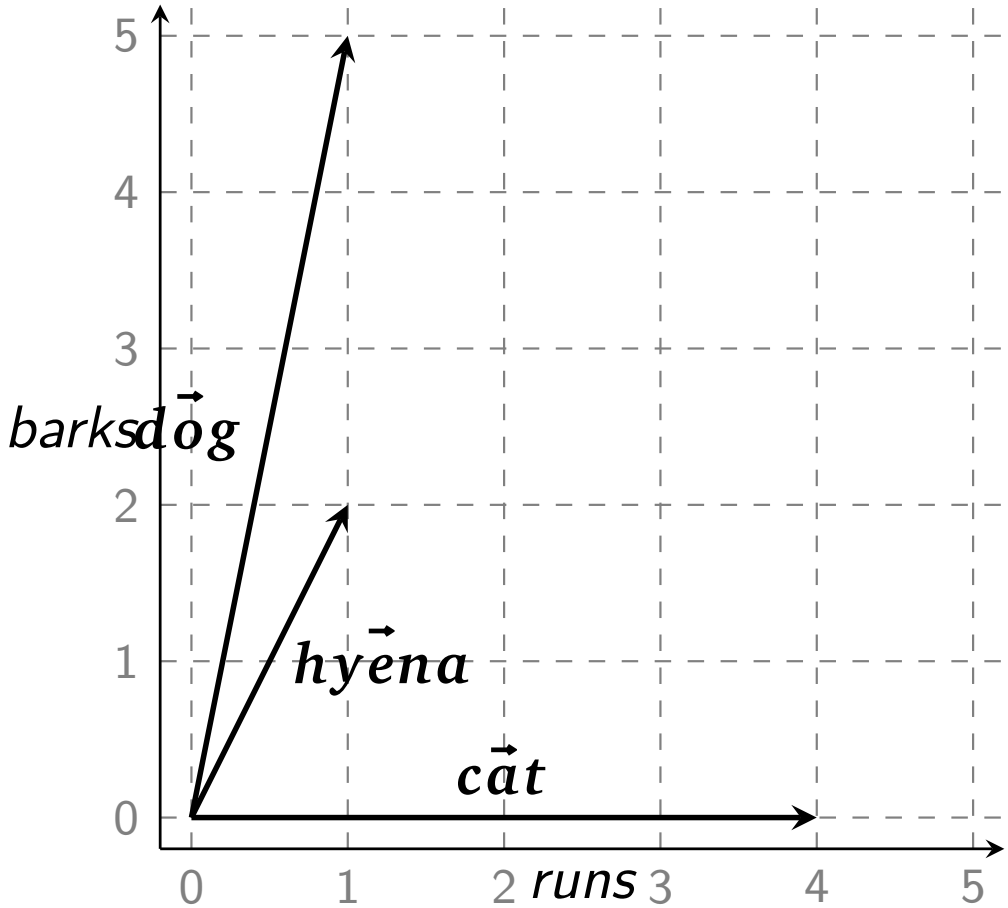
- Word embeddings correspond to the linguistic theory of "distributional semantics" (DS, or distributional semantics models, DSM)
- the general idea of both embeddings and DSM is that the meaning of a word can be known by the context in which it occurs, hence the quote from Firth (1957) : "You shall know a word by the company it keeps"
- Word embeddings are context-based vectorial representation of words used in machine learning, whereas DS is a semantic theory of meaning, which generally employs vectors to represent meanings.

Overview

"Word vectors"

from Baroni, Bernardi, and Zamparelli (2014)

	runs	*barks*
dog	1	5
hyena	1	2
cat	4	0



Overview

Why do word embeddings matter?

Word embeddings are widely used in descriptive & theoretical linguistics...

- **Sociolinguistics** Social bias (*Bolukbasi et al., 2016*)
- **Psycholinguistics** Mental cartography (*Louwerse and Zwaan, 2009*, *Louwerse and Benesh, 2012*), grounding (*Lenci, 2008*, *Lazaridou, Bruni, and Baroni, 2014*, *Kiela and Clark, 2015*), models for cognition (*Marelli and Baroni, 2015*)
- **Syntax** Syntactic/semantic composition (*Baroni, Bernardi, and Zamparelli, 2014*, *Mitchell and Lapata, 2008*, *Baroni and Zamparelli, 2010*, *Paperno et al., 2014*), external knowledge integration (*Mikolov et al., 2017*)
- **Semantics** Hypernym detection (*Roller, Erk, and Boleda, 2014*, *Santus et al., 2014*, *Santus et al., 2016*), mappings with formal semantics (*Herbelot and Vecchi, 2015*, *Kruszewski, Paperno, and Baroni, 2015*, *Kruszewski et al., 2016*)
- **Morphology** Morphological composition (*Marelli and Baroni, 2015*), theoretical morphology (*Bonami and Paperno, submitted*, *Varvara, 2017*, *Wauquier, 2017*)

Overview

Why do word embeddings matter?

They're also a key component of NLP, and especially of neural NLP, as they have a wide array of applications

- from predicting semantic relatedness judgments (Agirre et al., 2009)
- to initializing neural machine translation systems (Artetxe et al., 2017; Lample, Denoyer, and Ranzato, 2017).

Primer on linear algebra

Mathematical foundations of word embeddings

Primer on linear algebra

Vector spaces

Distributional semantics generally associate words from a given language with a vector space derived from their usage.

NB : the actual definition generally given is more lax: any model of word meaning that builds upon the idea that the meaning of a word can be retrieved from the context in which it occurs

The vector associated with a word will encode its contexts of occurrence.

Many algorithms have been adapted or conceived to achieve this (SVD, PCA, NMF, word2vec, GloVe, ELMo...)

Primer on linear algebra

Mathematical definition of vector spaces

From a mathematical point of view, a vector space V over a field F is a set possessing both

- an addition $V \times V \to V$ (two vectors added give a vector)
- and a scalar multiplication $F \times V \to V$ (the multiplication of a vector by a scalar is a vector).

Primer on linear algebra

Mathematical definition of vector spaces, remarks

- In DSMs, vectors must be associated with specific words or terms: two mathematically equivalent vector spaces where vectors are associated to different words should not receive the same interpretation
- in the case of NLP the field F almost always corresponds to the set of real numbers, whereas the vector space V itself is almost always defined as the set containing all possible sequences of real numbers of a given length d. In these cases, d is called the "dimension" of the vector space.
- in NLP, vectors are compared to one another mostly using offsets, distances or angles.

Exercises

Linear Algebra

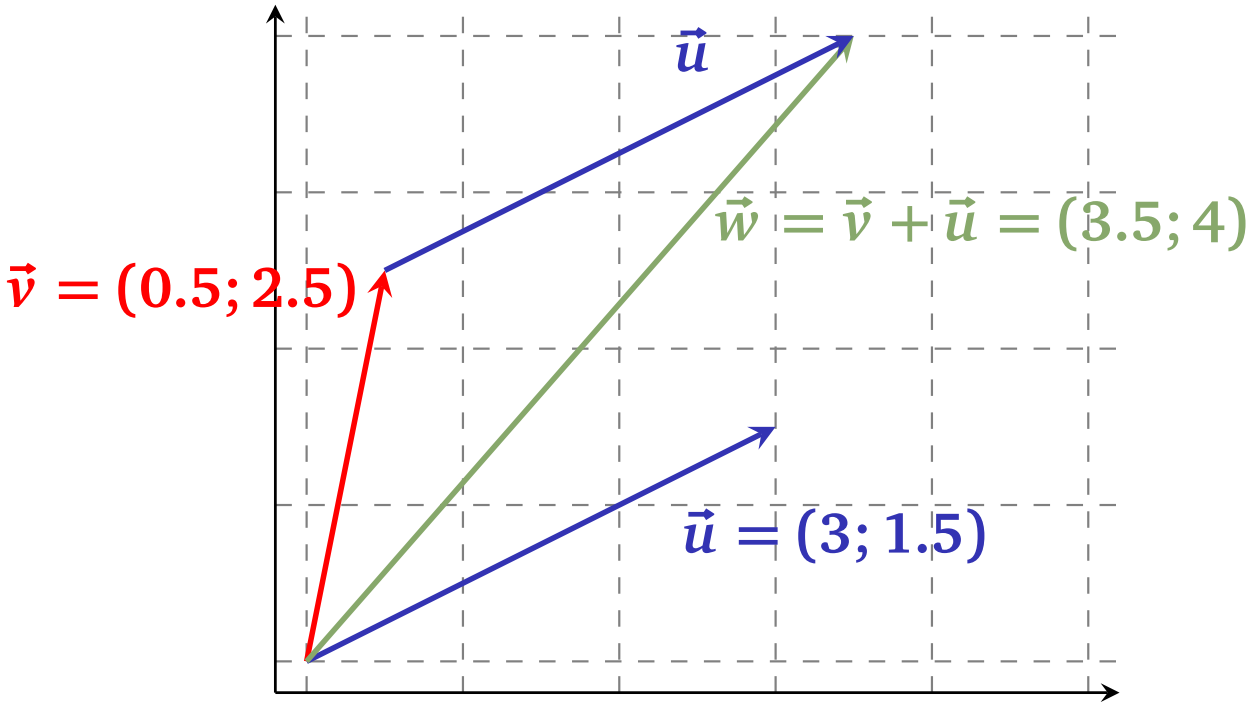
1. Implement a function checking whether two vectors have the same dimension.

   you can represent vectors as lists of float

Primer on linear algebra

Mathematical definition of vector spaces, vector addition

- In the traditional case, the addition of $\vec{v}$ and $\vec{u}$ is defined as the vector $\vec{w} = (v_1 + u_1; \ldots; v_d + u_d)$



- the addition possesses an identity element $\vec{0}$

$$\forall \vec{V} \ \ \vec{V} + \vec{0} = \vec{V}$$

$\vec{0}$ *corresponds to the sequence of length* d *composed only of 0.*

Primer on linear algebra

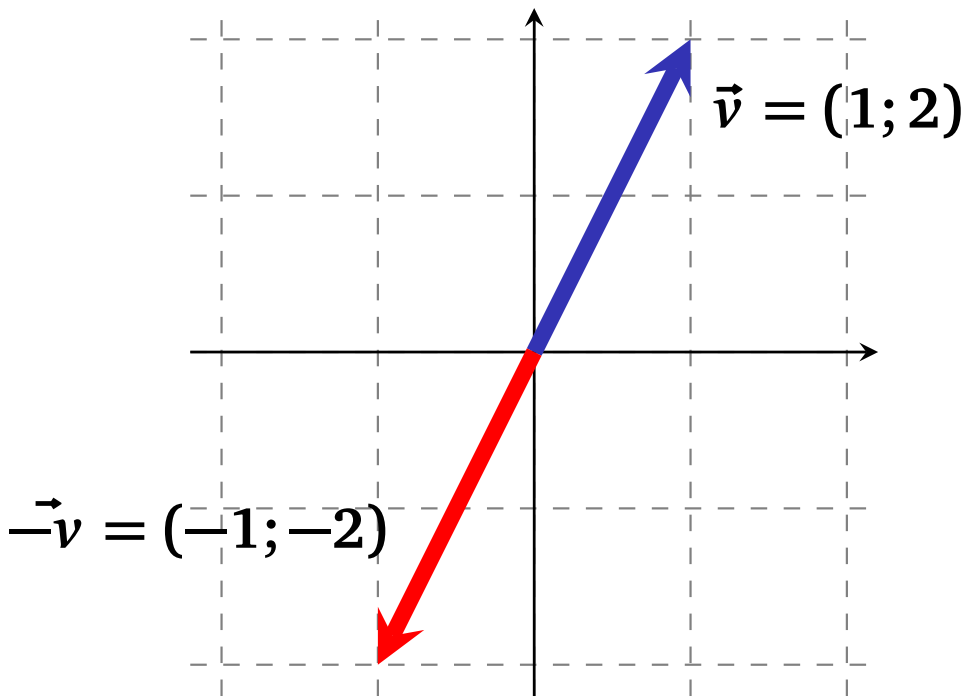
Mathematical definition of vector spaces, vector addition

- Inverse elements are defined for the addition:

$$\forall \vec{V} \ \exists \vec{V^{-1}} \text{ such as } \vec{V} + \vec{V^{-1}} = \vec{0}$$

  That is to say the addition of a vector and its inverse equals the identity element.

- In our case, the inverse of a vector $\vec{v}$ is the same vector except all of its components are of the opposite sign; it's more commonly written as $-\vec{v}$

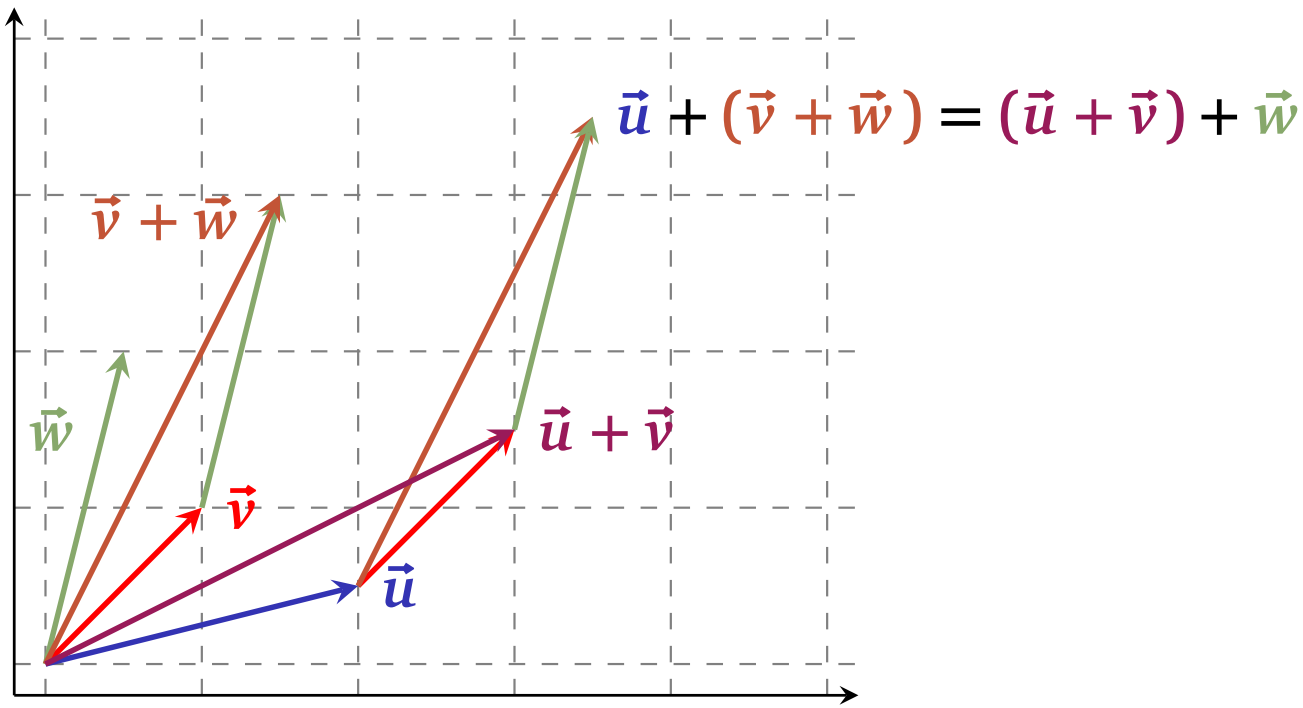


Primer on linear algebra

Mathematical definition of vector spaces, vector addition

- It is also associative ...

$$\vec{U} + (\vec{V} + \vec{W}) = (\vec{U} + \vec{V}) + \vec{W}$$



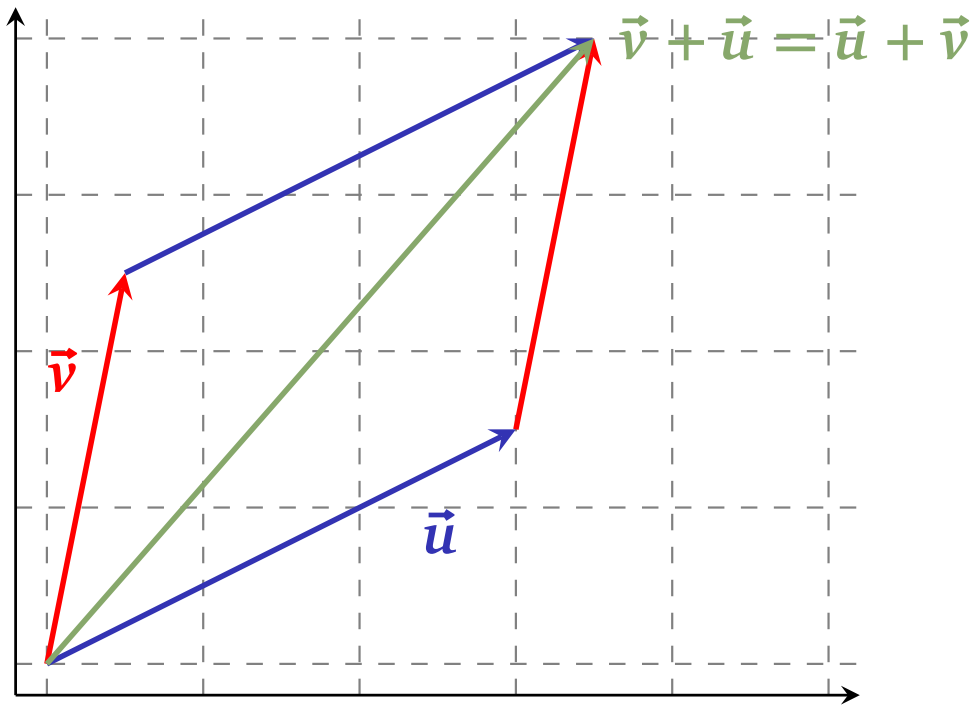
Primer on linear algebra

Mathematical definition of vector spaces, vector addition

- ... and commutative ...

$$\vec{V} + \vec{U} = \vec{U} + \vec{V}$$

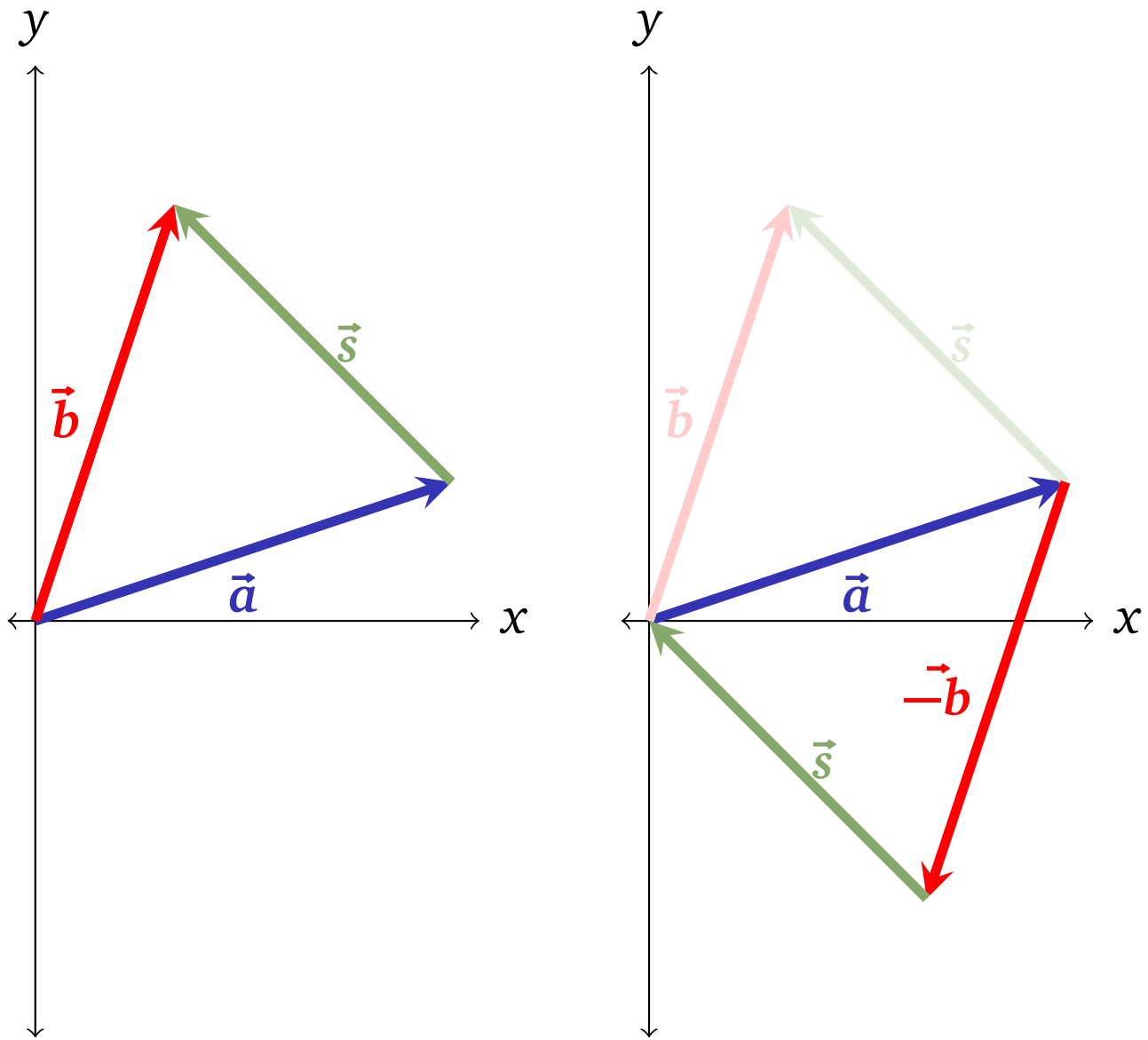
- commutativity and associativity imply that the order in which vectors are added has no impact on the result.



Primer on linear algebra

Vector offsets

The offset vector $\vec{s}$ corresponding to the "shift" from the vector $\vec{a}$ to the vector $\vec{b}$ can be computed by subtracting $\vec{b}$ from $\vec{a}$, which is equivalent to adding its inverse $-\vec{b}$.



Exercises

Linear Algebra

2. Implement vector addition as a function
3. Implement a function returning the inverse of a vector given as parameter
4. write a function that shows whether addition is commutative for two vectors given as parameters
5. write a function that shows whether addition is associative for three vectors given as parameters
6. write a function that returns the offset of two vectors given as parameters, ie. their difference

Primer on linear algebra

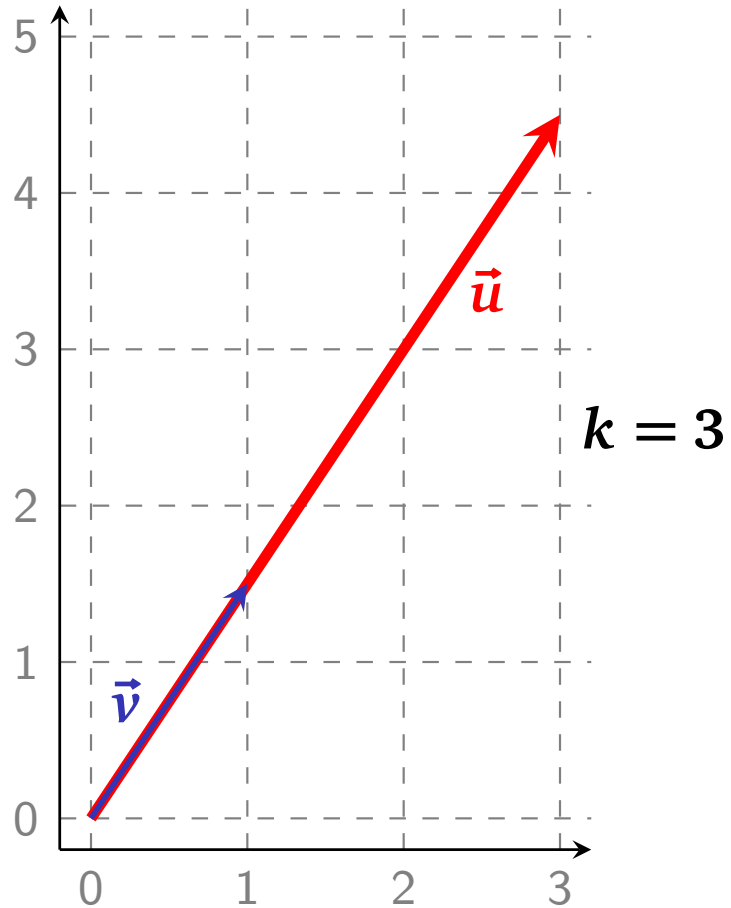
Scalar multiplication

- The scalar multiplication of vector $\vec{v}$ by scalar k is a vector $\vec{u}$ whose components are that of $\vec{v}$ multiplied by k.
- The scalar multiplication is compatible with the field multiplication:

$$(nm)\vec{V} = n(m\vec{V})$$

- the identity element of the field multiplication is the identity element for the scalar multiplication:

$$1\vec{V} = \vec{V}$$



Primer on linear algebra

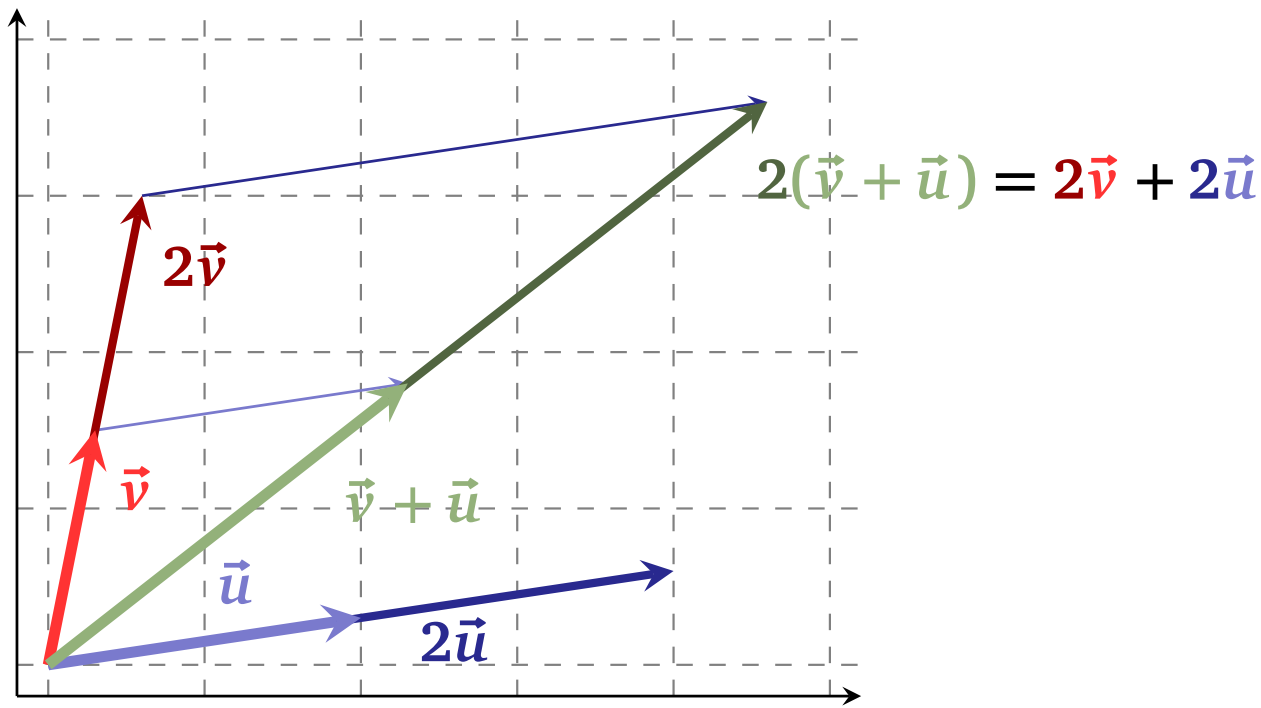
Mathematical definition of vector spaces, scalar multiplication

- the scalar multiplication is distributive over both field and vector additions:

$$n(\vec{V} + \vec{U}) = n\vec{V} + n\vec{U}$$

$$(n + m)\vec{V} = n\vec{V} + m\vec{V}$$

- $\vec{0}$ is generally defined as absorbing for the scalar multiplication.



Exercises

Linear Algebra

7. Implement scalar multiplication as a function
8. write a function that shows whether scalar multiplication is compatible with field (real numbers) multiplication for any two scalars and a vector given as parameters
9. write a function that shows whether scalar multiplication is distributive over field addition for any two scalars and a vector given as parameters
10. write a function that shows whether scalar multiplication is distributive over vectors addition for any two vectors and a scalar given as parameters

Primer on linear algebra

Scalar product

- The scalar product, or dot product, is a very frequently used vector operation defined on $V \times V \rightarrow F$
- It intuitively corresponds to how much two vectors go in the same direction
- It is defined as:

$$\vec{v} \cdot \vec{u} = \sum_{i=1}^{d} v_i \times u_i$$

Exercises

Linear Algebra

11. Implement scalar product as a function

Primer on linear algebra

Euclidean distance and norm

- the Euclidean distance between two vectors can be computed with the formula:

$$\text{distance}(\vec{v}, \vec{u}) = \sqrt{\sum_{i=1}^{d} |v_i - u_i|^2}$$

- the Euclidean norm of a vector (informally, its "length") is given by:

$$|\vec{v}| = \sqrt{\sum_{i=1}^{d} |v_i|^2}$$

- It is therefore equivalent to the distance to the origin $\vec{0}$; likewise, the distance between two vectors corresponds to the norm of the vector offset.

Exercises

Linear Algebra

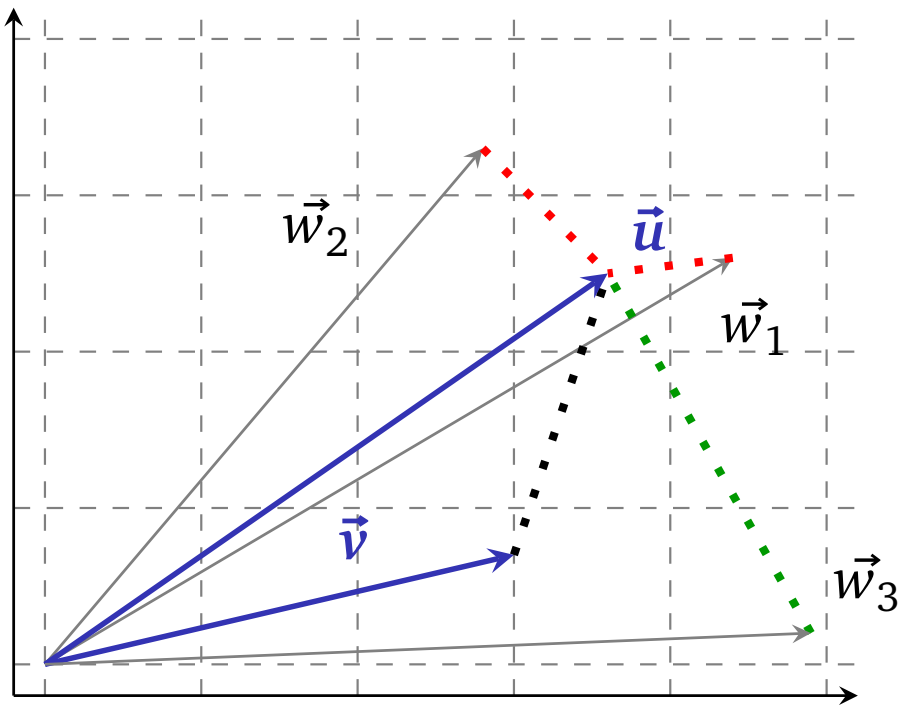
12. Implement euclidean distance as a function
13. Implement euclidean norm as a function

Primer on linear algebra

Rank measurements

- To take into account the fact that some word vectors are clustered together, we sometimes use rank to measure whether a vector $\vec{v}$ is among the closest neighbors of a vector $\vec{u}$:

$$\text{rank}(\vec{v},\vec{u}) = \#\{\vec{w} \mid \text{distance}(\vec{w},\vec{u}) > \text{distance}(\vec{v},\vec{u}) \ \wedge \ \vec{w} \in V \setminus \{\vec{u}\}\} + 1$$



Exercises

Linear Algebra

14. Implement rank as a function

Primer on linear algebra

Cosine similarity

- the cosine of the angle θ given by two vectors $\vec{v}$ and $\vec{u}$ can be computed as:

$$\cos(\theta) = \cos(\vec{v}, \vec{u}) = \frac{\vec{v} \cdot \vec{u}}{|\vec{v}||\vec{u}|} = \frac{\sum_{i=1}^{d} v_i \times u_i}{\sqrt{\sum_{i=1}^{d} v_i^2} \times \sqrt{\sum_{i=1}^{d} u_i^2}}$$

- if two vectors are equal, their cosine is equal to 1:

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

- Hence, the cosine can be and often is used as a similarity measure.

Primer on linear algebra

Cosine similarity, caveat

- However if $\vec{u} = k\vec{v}$, then $\cos(\vec{v}, \vec{u}) = 1$:

$$\cos(\vec{v}, k\vec{v}) = \frac{\vec{v} \cdot k\vec{v}}{|\vec{v}||k\vec{v}|} = \frac{\sum_{i=1}^{d} v_i^2 \times k}{\sqrt{\sum_{i=1}^{d} v_i^2} \times \sqrt{\sum_{i=1}^{d} (v_i \times k)^2}}$$

$$= \frac{k \times \sum_{i=1}^{d} v_i^2}{\sqrt{\sum_{i=1}^{d} v_i^2} \times \sqrt{k^2 \sum_{i=1}^{d} v_i^2}} = \frac{k \times \sum_{i=1}^{d} v_i^2}{k \times \sum_{i=1}^{d} v_i^2} = 1$$

- In the case of DSMs, vectors are often (but not always) normalized, in which case $\cos(\vec{v}, \vec{u}) = 1 \iff \vec{v} = \vec{u}$.

Exercises

Linear Algebra

15. Implement cosine similarity as a function

    you can use the functions you previously implemented

16. Homework (due Tuesday noon) : euclidean distance is also called "l_2 norm". It is a specific case of a family of distance measures, called "p-distances", where $p = 2$, because we take the **square** root and sum the the difference between the components **squared**. Implement a function that, for any p and pair of vectors passed as parameters, returns the relevant p-distance between the two vectors.

$$p\text{-distance}(\vec{v}, \vec{u}) = \sqrt[p]{\sum_{i=1}^{d} |v_i - u_i|^p}$$

    don't hesitate to hand in partial solutions: try implementing the function with a fixed value for p first

17. Homework (due Tuesday noon) : the euclidean norm is to the euclidean distances as the p-norms to the p-distances. Implement a function returning the p-norm of a vector for any value of p and any vector given as parameters.

$$|\vec{v}|_p = \sqrt[p]{\sum_{i=1}^{d} |v_i|^p}$$

Applications

Applications

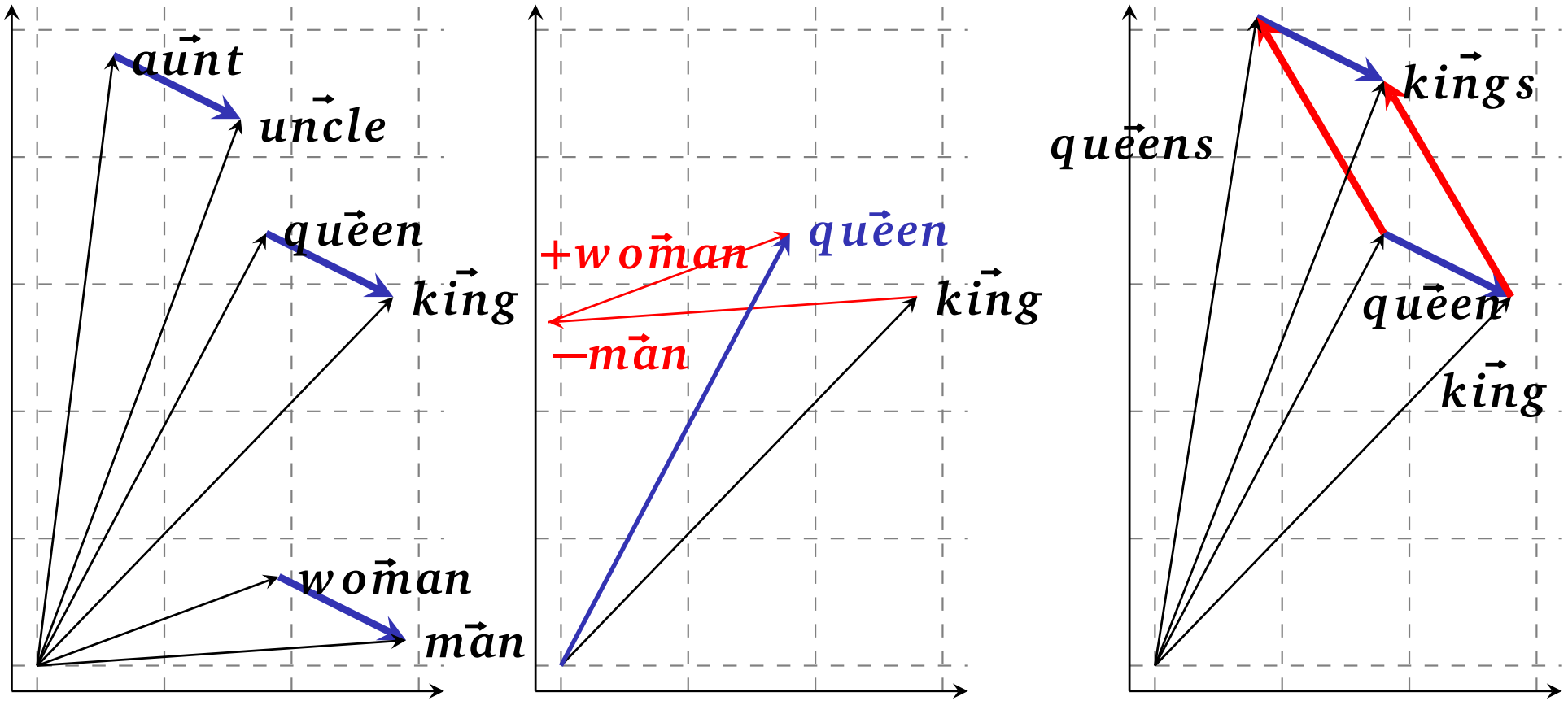
Main usages

- Word embeddings are most often used to compute the similarity of two words, using a similarity function between the two associated vectors.
- A natural extension consists in selecting words similar to a given (word) vector
- This approach has been explored most famously in formal analogy operationalization (Mikolov, Yih, and Zweig, 2013)
- The class will be focused on reproducing the setup step-by-step

Applications

Formal analogy

Using vector addition, Mikolov, Yih, and Zweig (2013) tried to operationalize formal analogy.



Exercises

Formal analogy step by step

18. download the word embeddings sample (from fastText) from the github for this class: https://github.com/TimotheeMickus/lexical_resources; install numpy using pip if it's not installed
19. the word2vec sample has the following format:
    - the first line contains the number of word embeddings, followed by a space, followed by the dimension of the vectors
    - every line excepted the first contains a word, then a space, then the list of components of the associated vectors separated by spaces.

    write a function that:
    - constructs a numpy matrix of the shape $V \times d$, where V is the number of embeddings and d the dimension
    - fills this matrix with the information from the word2vec sample
    - returns both the matrix and a "lookup dictionary", which associates words to the index of their respective embeddings in the matrix

Exercises

Formal analogy step by step

20. write a function taking the embedding matrix, the lookup dictionary and a word as parameters, and returning the embedding corresponding to the word.
21. The additive formal analogy operationalization supposes that: if *queen* is to *king* as *woman* is to *man*, then $\vec{king} - \vec{man} \approx \vec{queen} - \vec{woman}$, and therefore that $\vec{queen} \approx \vec{king} - \vec{man} + \vec{woman}$. Write a function that given a pair of words instantiating a relation (eg. "man" and "woman") and a cue word (eg. "king") computes the expected vector (eg. the approximation for $\vec{queen}$).
22. write a function that returns the similarity score between a word and a given vector passed as parameters: use the lookup dictionary to retrieve the embedding for the word, and cosine as a similarity measure between the vectors.

Exercises

Formal analogy step by step

23. Retrieve the formal analogy dataset from the github for this class : https://github.com/TimotheeMickus/lexical_resources (copied from tensorflow).
24. The dataset has the following format:
    - a section starts with a header, beginning with a colon ":" followed by the name of a relation
    - below the header of a function are the relevant examples: each examples is a line containing four words, such as the two first words should have the same relation to one another as the two last words : eg. "king queen man woman"

    write a function that reads this dataset and separates it into a list of sections. Each section should be a list of examples, and each example should be a list of four words.
25. write a function that takes a list of examples, computes the expected vectors using the first word as a cue and the last two words as the instance of the relation (use exercise 21), and then computes the similarity between the expected vector and the second word (use exercise 22).
26. compute the average similarity across all examples. What do you think?

Exercises

Formal analogy step by step

27. Homework : Other functions can be used to assess similarity. Try adapting the function written in exercise 22 so that it accepts a similarity measure as a parameter. Try your function using euclidean distance and rank. How does this change the results you get in exercise 25 ?
28. Homework : another very famous similarity measure is the "cosmul" similarity (Levy and Goldberg, 2014). When testing whether b_2 (eg. $\vec{queen}$) is to b_1 (eg. $\vec{king}$) as a_2 (eg. $\vec{woman}$) is to a_1 (eg. $\vec{man}$), it is defined as:

$$\frac{\cos(b_2, b_1)\cos(b_2, a_2)}{\cos(b_2, a_1) + \epsilon}$$

The ϵ here is used to ensure no divisions by zero are made. The original paper used $\epsilon = 0.001$. Implement the cosmul similarity function, and use it in the function written in exercise 22 instead of cosine.

29. Homework : compute and compare macro and micro average across the sections of the dataset downloaded in 23. What can you infer?