92586 Computational Linguistics

Lesson 2. Vector Space Model

Alberto Barrón-Cedeño

Alma Mater Studiorum-Università di Bologna
a.barron@unibo.it @_albarron_

20/02/2020

Table of Contents

1. Current Situation
2. Representations Revisited
3. Sentiment Analysis

Current Situation

Current Situation

What you know

- What is computational linguistics / natural language processing

What you have done (on your own)

What you can do

Current Situation

What you know

- What is computational linguistics / natural language processing
- There are two main paradigms: rule-based and statistical

What you have done (on your own)

What you can do

Current Situation

What you know

- What is computational linguistics / natural language processing
- There are two main paradigms: rule-based and statistical
- How to identify the rule-based paradigm

What you have done (on your own)

What you can do

Current Situation

What you know

- What is computational linguistics / natural language processing
- There are two main paradigms: rule-based and statistical
- How to identify the rule-based paradigm

What you have done (on your own)

- You have setup a Python development environment

What you can do

Current Situation

What you know

- What is computational linguistics / natural language processing
- There are two main paradigms: rule-based and statistical
- How to identify the rule-based paradigm

What you have done (on your own)

- You have setup a Python development environment
  1. PyCharm (or any other option, e.g., Eclipse)
  2. Google's Colab

What you can do

Current Situation

What you know

- What is computational linguistics / natural language processing
- There are two main paradigms: rule-based and statistical
- How to identify the rule-based paradigm

What you have done (on your own)

- You have setup a Python development environment
  1. PyCharm (or any other option, e.g., Eclipse)
  2. Google's Colab
- You studied **git** and LaTeX

What you can do

Current Situation

What you know

- What is computational linguistics / natural language processing
- There are two main paradigms: rule-based and statistical
- How to identify the rule-based paradigm

What you have done (on your own)

- You have setup a Python development environment
  1. PyCharm (or any other option, e.g., Eclipse)
  2. Google's Colab
- You studied **git** and LaTeX
- You forked your project template from Github

What you can do

Current Situation

What you know

- What is computational linguistics / natural language processing
- There are two main paradigms: rule-based and statistical
- How to identify the rule-based paradigm

What you have done (on your own)

- You have setup a Python development environment
  1. PyCharm (or any other option, e.g., Eclipse)
  2. Google's Colab
- You studied **git** and LaTeX
- You forked your project template from Github

What you can do

- Open a text file

Current Situation

What you know

- What is computational linguistics / natural language processing
- There are two main paradigms: rule-based and statistical
- How to identify the rule-based paradigm

What you have done (on your own)

- You have setup a Python development environment
  1. PyCharm (or any other option, e.g., Eclipse)
  2. Google's Colab
- You studied **git** and LaTeX
- You forked your project template from Github

What you can do

- Open a text file
- Tokenise and extract *n*-grams

Current Situation

What you know

- What is computational linguistics / natural language processing
- There are two main paradigms: rule-based and statistical
- How to identify the rule-based paradigm

What you have done (on your own)

- You have setup a Python development environment
  1. PyCharm (or any other option, e.g., Eclipse)
  2. Google's Colab
- You studied **git** and LaTeX
- You forked your project template from Github

What you can do

- Open a text file
- Tokenise and extract *n*-grams
- Normalise text

Current Situation

What you know

- What is computational linguistics / natural language processing
- There are two main paradigms: rule-based and statistical
- How to identify the rule-based paradigm

What you have done (on your own)

- You have setup a Python development environment
  1. PyCharm (or any other option, e.g., Eclipse)
  2. Google's Colab
- You studied **git** and LaTeX
- You forked your project template from Github

What you can do

- Open a text file
- Tokenise and extract *n*-grams
- Normalise text
- Build document representations

Current Situation

What you know

- What is computational linguistics / natural language processing
- There are two main paradigms: rule-based and statistical
- How to identify the rule-based paradigm

What you have done (on your own)

- You have setup a Python development environment
  1. PyCharm (or any other option, e.g., Eclipse)
  2. Google's Colab
- You studied **git** and LaTeX
- You forked your project template from Github

What you can do

- Open a text file
- Tokenise and extract *n*-grams
- Normalise text
- Build document representations
- Find out alternatives if not working in English

Representations Revisited

Representations Revisited

[1] http://www.nltk.org/

Representations Revisited

1. Using NLTK[1] to tokenize

[1] http://www.nltk.org/

Representations Revisited

1. Using NLTK[1] to tokenize
2. Using `.lower()` to ignore capitalisation

[1] http://www.nltk.org/

Representations Revisited

1. Using NLTK[1] to tokenize
2. Using `.lower()` to ignore capitalisation
3. Using Porter's stemmer to drop suffixes

[1] http://www.nltk.org/

Representations Revisited

1. Using NLTK[1] to tokenize
2. Using `.lower()` to ignore capitalisation
3. Using Porter's stemmer to drop suffixes
4. Using a lemmatiser to find the root of the words

[1] http://www.nltk.org/

Representations Revisited

1. Using NLTK[1] to tokenize
2. Using `.lower()` to ignore capitalisation
3. Using Porter's stemmer to drop suffixes
4. Using a lemmatiser to find the root of the words
5. **Discarding stopwords from the text**

[1] http://www.nltk.org/

Representations Revisited

1. Using NLTK[1] to tokenize
2. Using `.lower()` to ignore capitalisation
3. Using Porter's stemmer to drop suffixes
4. Using a lemmatiser to find the root of the words
5. **Discarding stopwords from the text**
6. **Building a vectorial representation**

[1] http://www.nltk.org/

Stopwords

Common words in a language that occur with a high frequency but carry much less substantive information about the meaning of a phrase (Lane et al., 2019, p. 51–54)

[2] For instance, from NLTK, sklearn, or `https://github.com/stopwords-iso`

Stopwords

Common words in a language that occur with a high frequency but carry much less substantive information about the meaning of a phrase (Lane et al., 2019, p. 51–54)

Option 1 Consider the most frequent tokens in a reference corpus as stopwords (remember Genesis?)

[2] For instance, from NLTK, sklearn, or `https://github.com/stopwords-iso`

Stopwords

Common words in a language that occur with a high frequency but carry much less substantive information about the meaning of a phrase (Lane et al., 2019, p. 51–54)

Option 1 Consider the most frequent tokens in a reference corpus as stopwords (remember Genesis?)

Option 2 Take a an existing list of stopwords[2]

en	**es**	**it**
i	a	altri
me	ahora	certa
my	alli	della
it	cerca	nessuna
is	el	prima
do	es	quello
the	unas	solito
will	vez	va
other	yo	via

[2] For instance, from NLTK, sklearn, or `https://github.com/stopwords-iso`

Stopwords

Discarding stopwords

- They are the most frequent tokens in the documents
- Discarding them reduces the computational effort significantly

Stopwords

Discarding stopwords

- They are the most frequent tokens in the documents
- Discarding them reduces the computational effort significantly
- Typical size of a stopwords list: a few hundred words
- For some applications (e.g., **topic clustering**), they can be safely discarded
- For some others (e.g., **dialogue**) they cannot

Stopwords

Discarding stopwords

- They are the most frequent tokens in the documents
- Discarding them reduces the computational effort significantly
- Typical size of a stopwords list: a few hundred words
- For some applications (e.g., **topic clustering**), they can be safely discarded
- For some others (e.g., **dialogue**) they cannot

Stopwords have to be considered with a grain of salt (as most in NLP)

Vector representation

BoW

- A text is represented as the bag (multiset) of its words
- It disregards grammar
- It disregards word order
- It (can) consider frequency

From (Lane et al., 2019, p. 41)

Dot product

Algebraically, it is the sum of the products of the corresponding entries of the two sequences of numbers $a \cdot b$

$$a \cdot b = \sum_{i=1}^{n} a_i b_i$$

Dot product

Algebraically, it is the sum of the products of the corresponding entries of the two sequences of numbers $a \cdot b$

$$
\begin{aligned}
a \cdot b &= \sum_{i=1}^{n} a_i b_i \\
&= a_1 b_1 + a_2 b_2 + a_3 b_3 + \ldots a_n b_n
\end{aligned}
$$

Dot product

Algebraically, it is the sum of the products of the corresponding entries of the two sequences of numbers $a \cdot b$

$$
\begin{aligned}
a \cdot b &= \sum_{i=1}^{n} a_i b_i \\
&= a_1 b_1 + a_2 b_2 + a_3 b_3 + \ldots a_n b_n
\end{aligned}
$$

```
v1 = [1,2,3]
v2 = [3,4,6]
my_sum = 0
for i in range(len(v1)):
    my_sum += v1[i] * v2[i]
```

(there are better —more efficient— ways to compute the dot product)

Dot product

Algebraically, it is the sum of the products of the corresponding entries of the two sequences of numbers $a \cdot b$

$$
\begin{aligned}
a \cdot b &= \sum_{i=1}^{n} a_i b_i \\
&= a_1 b_1 + a_2 b_2 + a_3 b_3 + \ldots a_n b_n
\end{aligned}
$$

```
v1 = [1,2,3]
v2 = [3,4,6]
my_sum = 0
for i in range(len(v1)):
    my_sum += v1[i] * v2[i]
```

(there are better —more efficient— ways to compute the dot product)

Now, we can use the dot product to compare two documents ($\sim$ similarity)

Vector space model

We just built our first vector space model!
"[...] an **algebraic** model for representing text documents (and any objects, in general) as vectors of identifiers [...]"[3]

[3] https://en.wikipedia.org/wiki/Vector_space_model

Vector space model

We just built our first vector space model!
"[...] an **algebraic** model for representing text documents (and any objects, in general) as vectors of identifiers [...]"[3]

Some applications

- Relevance rankings in keyword-based search
- Text clustering to "discover" structure and relations in a text collection
- Reading recommendations

(Not the SoA for most tasks, but it's a starting point)

[3] https://en.wikipedia.org/wiki/Vector_space_model

Sentiment Analysis

Sentiment Analysis

It **does not** refer to real sentiment, such as love or hate

It is about **positive** and **negative** (and **neutral**)

From (Lane et al., 2019, p. 62–65)

Sentiment Analysis

It **does not** refer to real sentiment, such as love or hate

It is about **positive** and **negative** (and **neutral**)

This monitor is definitely a good value. Does it have superb color and contrast? No. Does it boast the best refresh rate on the market? No. But if you're tight on money, this thing looks and preforms great for the money. It has a Matte screen which does a great job at eliminating glare. The chassis it's enclosed within is absolutely stunning.

From (Lane et al., 2019, p. 62–65)

Sentiment Analysis

It **does not** refer to real sentiment, such as love or hate

It is about **positive** and **negative** (and **neutral**)

This monitor is definitely a good value. Does it have superb color and contrast? No. Does it boast the best refresh rate on the market? No. But if you're tight on money, this thing looks and preforms great for the money. It has a Matte screen which does a great job at eliminating glare. The chassis it's enclosed within is absolutely stunning.

POSITIVE

From (Lane et al., 2019, p. 62–65)

Sentiment Analysis

It **does not** refer to real sentiment, such as love or hate

It is about **positive** and **negative** (and **neutral**)

This monitor is definitely a good value. Does it have superb color and contrast? No. Does it boast the best refresh rate on the market? No. But if you're tight on money, this thing looks and preforms great for the money. It has a Matte screen which does a great job at eliminating glare. The chassis it's enclosed within is absolutely stunning.

POSITIVE

His [ssa] didnt concede until July 12, 2016. Because he was throwing a tantrum. I can't say this enough: [kcuF] Bernie Sanders.

From (Lane et al., 2019, p. 62–65)

Sentiment Analysis

It **does not** refer to real sentiment, such as love or hate

It is about **positive** and **negative** (and **neutral**)

This monitor is definitely a good value. Does it have superb color and contrast? No. Does it boast the best refresh rate on the market? No. But if you're tight on money, this thing looks and preforms great for the money. It has a Matte screen which does a great job at eliminating glare. The chassis it's enclosed within is absolutely stunning.

POSITIVE

His [ssa] didnt concede until July 12, 2016. Because he was throwing a tantrum. I can't say this enough: [kcuF] Bernie Sanders.

NEGATIVE

From (Lane et al., 2019, p. 62–65)

Sentiment Analysis

VADER a rule-based approach

Valence **A**ware **D**ictionary for s**E**ntiment **R**easoning[4]

[4] http://comp.social.gatech.edu/papers/icwsm14.vader.hutto.pdf

Sentiment Analysis

VADER a rule-based approach

Valence **A**ware **D**ictionary for s**E**ntiment **R**easoning[4]

- It has a lexicon packed with tokens and their associated "sentiment" score
- It counts all tokens belonging to each category: [positive, neutral, negative]

[4] http://comp.social.gatech.edu/papers/icwsm14.vader.hutto.pdf

Valence **A**ware **D**ictionary for s**E**ntiment **R**easoning[5]

[5] http://comp.social.gatech.edu/papers/icwsm14.vader.hutto.pdf

Valence **A**ware **D**ictionary for s**E**ntiment **R**easoning[5]

- It has a lexicon packed with tokens and their associated "sentiment" score
- It counts all tokens belonging to each category: [pos, neu, neg]...
- ...and combine them to determine the sentiment

[5] http://comp.social.gatech.edu/papers/icwsm14.vader.hutto.pdf

Coming soon...

Statistical NLP

References

Lane, H., C. Howard, and H. Hapkem
2019. *Natural Language Processing in Action*. Shelter Island, NY: Manning Publication Co.