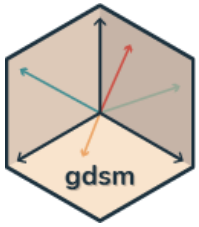

Session 05: fastText

Viktoria Schneider & Dominic Schmitz

Verein für Diversität in der Linguistik

Was ist fastText?

- FastText (Bojanowski 2016) ist ein Überbegriff für verschiedene Dinge
  - Hier zu finden: https://fasttext.cc/

1. **Package**, Python, inzwischen wohl auch R
2. **Vektorräume** mit vortrainierten Daten für verschieden Sprachen
3. **Möglichkeit** anhand dieser vortrainierten Daten eigene Vektoren zu berechnen (= fastText Model)

1. Package, Python

- Folgendes Package muss geladen werden:
  - pip install fasttext (https://pypi.org/project/fasttext/)
- Voraussetzungen
  - C++11 compiler,
  - Python (version 2.7 or ≥ 3.4),
  - NumPy & SciPy & pybind11

1. Package, Python

- Dann importiert ihr die Packages

```
#install following packages first
import gzip
import gensim
import numpy as np
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
plt.rcParams['figure.figsize'] = [15, 7.5]
plt.rcParams.update({'font.size': 20})
import pandas as pd
import fasttext
import torch

from numpy import genfromtxt
from numpy import dot
from numpy.linalg import norm
```

1. Package, Python

- Dann installiert ihr *fastText* (hier *CommonCrawl* subword Model English)

```
import fasttext
import io


def load_vectors(VScrawlsub):
    fin = io.open(VScrawlsub, 'r', encoding='utf-8', newline='\n',
errors='ignore')
    n, d = map(int, fin.readline().split())
    data = {}
    for line in fin:
        tokens = line.rstrip().split(' ')
        data[tokens[0]] = map(float, tokens[1:])
    return data
modelft = fasttext.load_model(r“yourPath\crawl-300d-2M-subword\crawl-300d-2M-
subword.bin “)
```

Das ist euer Model, ihr könnt das nennen, wie ihr wollt.

Erinnerung

CBOW

- Vorkommnisse jedes einzelnen Wortes im Vektorraum mit jedem anderen Wort im Vektorraum
- Dimensionen reduziert → je nach Forschungsfrage 100 – unendlich (Wörter im Vektorraum)

	Geld	Institut	Sitz	Park
Bank.1	35	26	15	0
Bank.2	0	0	37	60

Erinnerung

Skip-Gram

- Vorkommnisse jedes einzelnen Wortes im Vektorraum mit jedem anderen Wort im Vektorraum und deren *n*-grams
  - *n*-grams: für Deutsch und Englisch sind 3-6-grams sinnvoll (Bojanoskwxxx)
- Dimensionen reduziert → je nach Forschungsfrage 100 – unendlich (Wörter im Vektorraum)

	#ba	ban	ank	nk#
Bank	1	1	1	1
Bar	1	0	0	0

2. Vektorräume mit vortrainierten Daten

- Es gibt vortrainierte Vektorräume
  - https://fasttext.cc/docs/en/english-vectors.html (English)
  - CBOW: *Wikipedia*, *CommonCrawl*
  - Skip-gram: *Wikipedia*-subword, *CommonCrawl*-subword
  - Modelle in beiden Varianten für 157 weitere Sprachen
    - https://fasttext.cc/docs/en/crawl-vectors.html

3. Eigene Vektoren zu berechnen

- Mit dem Model aus 1 könnt ihr eigene Wort-Listen laden zum Berechnen von Vektoren, Cosinus-Ähnlichkeiten, nächste Nachbarn, etc.
- Ihr könnt die Vektoren dann auch exportieren und bspw. in R laden

```
#list of vectors for DSM Workshop
DSMW = genfromtxt(r"E:\DSM-Workshop\word_list.txt", delimiter=';', dtype =
None, encoding='utf-8')


print(DSMW)
list_of_vectors_DSMW = [modelft.get_word_vector(x) for x in DSMW]


print(list_of_vectors_DSMW )


pd.DataFrame(list_of_vectors_DSMW).to_csv("DSM-workshop-vectors.csv",
header=None, index=None)
```