Document Similarity Benchmark

Jan-Gabriel Mylius

Lehrstuhl für Datenbanksysteme,
Institut für Informatik,
Ruprecht-Karls-Universität Heidelberg
mylius@posteo.de

Vorlage DBS
10. April 2018, Heidelberg

Motivation

- NLP is often interested in meaning of a sentence.
- Comparing the semantic similarity is a common task

Sentence 1	Sentence 1
The man is operating a stenograph.	The man is typing on a machine used for stenography.
”Fairies don’t exist” – fine.	”Satyrs don’t exist” – fine.

We want a framework:

- That compares algorithms on different datasets.
- Is easily expandable.
- Provides commonly used algorithms.
- Provides common datasets.

The Architecture

The benchmark suite consists of two essential elements:

- the algorithms
- the datasets

For both of these we want to have a base class that allows easy instantiation of customized versions.

Algorithms

There is a template class for implementing custom algorithms. The following functions need to be implemented:

- train(self, in_dataset, in_score)
- encode(self, in_line)
- compare(self, a, b)

Preimplemented Algorithms

The benchmark suit offers a number of already implemented algorithms:

- Bag of Words
- Bag of Words with lemmatization
- Word2Vec[Mik+15]
- BERT[Dev+18]
- Word Mover's Distance[Kus+15]
- Doc2Vec[LM14]

For some algorithms several similarity metrics where implmented:

- Consine similarity
- Euclidian similarity
- Jaccard similarity

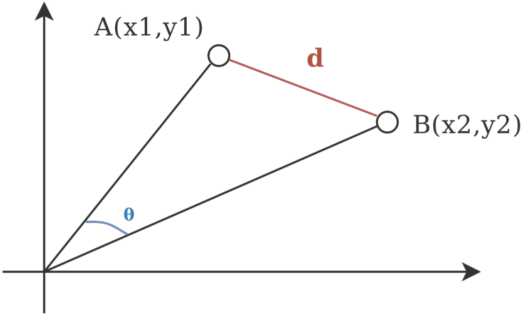
Functional Principle

How do these measures function:

1. Convert Words to Vector
2. Compare vectors using a similarity metric

For some algorithms several similarity metrics were implemented:

- Consine similarity
- Euclidian similarity
- Jaccard similarity



Euclidean vs Cosine Distance
Source:
Chris Emmery [Emm17]

The dataset class

The dataset class has the following essential functions:

- load(self, path, data_cols, data, score_col, scores, id_col=None, ids=None)
- load_sts(self)
- load_sick(self)
- calc_results(self, alg)
- compare(self, function, alg)
- miscellaneous internal functions

What it does:

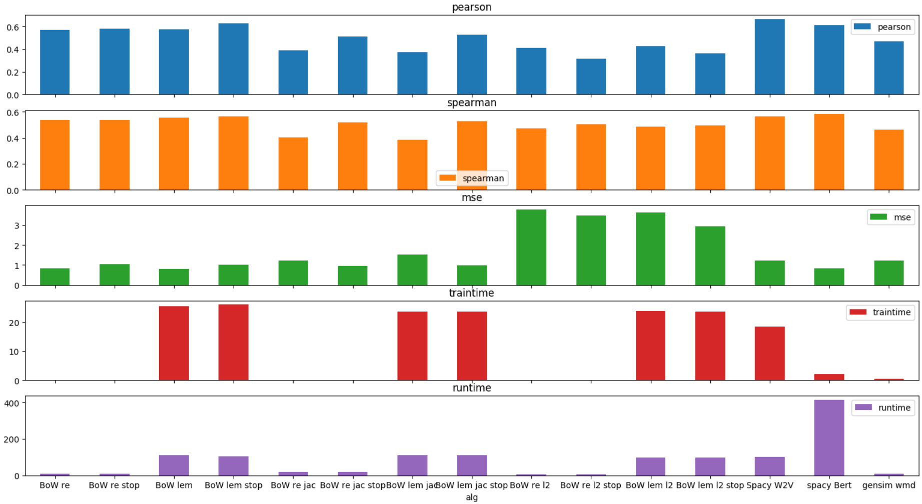
- loads a dataset
- runs an algorithm
- norms results to match the datasets scale
- stores results for all algorithms that have run

The benchmark

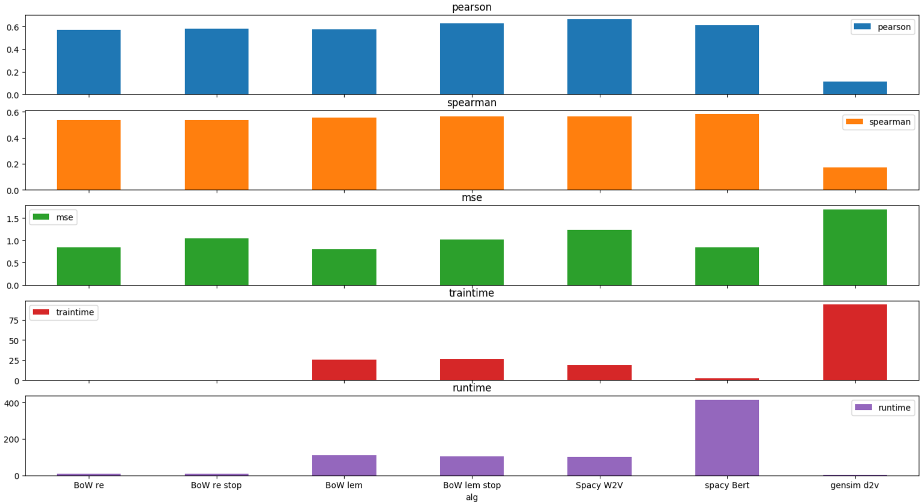
What the benchmark does:

- Accepts commandline parameters
  - No parameters: Run all algorithms
  - Else: Run given algorithms
- Measures:
  - Training Time
  - Run Time
  - Pearson correlation
  - Spearman's correlation
  - Mean squared error
- Outputs results to json

Overview of all results



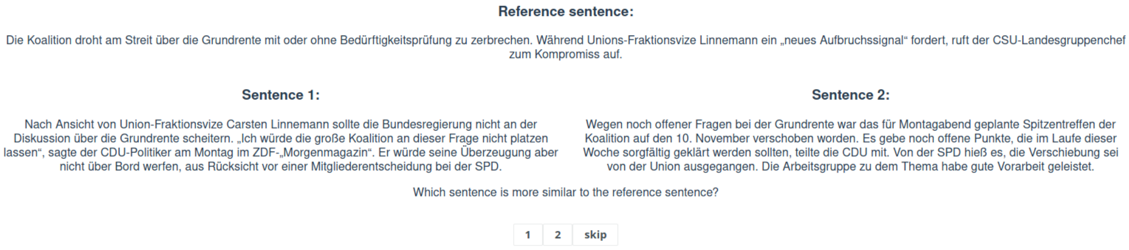
Overview of results without l2 and jaccard



Discussion of Results

Lessons Learned

The interface.



Reference sentence:

Die Koalition droht am Streit über die Grundrente mit oder ohne Bedürftigkeitsprüfung zu zerbrechen. Während Unions-Fraktionsvize Linnemann ein „neues Aufbruchssignal“ fordert, ruft der CSU-Landesgruppenchef zum Kompromiss auf.

Sentence 1:

Nach Ansicht von Union-Fraktionsvize Carsten Linnemann sollte die Bundesregierung nicht an der Diskussion über die Grundrente scheitern. „Ich würde die große Koalition an dieser Frage nicht platzen lassen“, sagte der CDU-Politiker am Montag im ZDF-„Morgenmagazin“. Er würde seine Überzeugung aber nicht über Bord werfen, aus Rücksicht vor einer Mitgliederentscheidung bei der SPD.

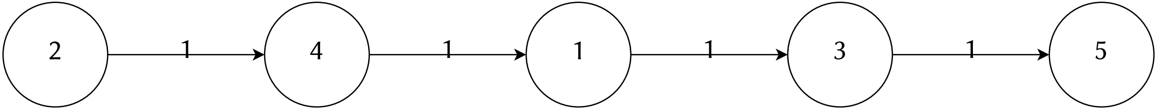
Sentence 2:

Wegen noch offener Fragen bei der Grundrente war das für Montagabend geplante Spitzentreffen der Koalition auf den 10. November verschoben worden. Es gebe noch offene Punkte, die im Laufe dieser Woche sorgfältig geklärt werden sollten, teilte die CDU mit. Von der SPD hieß es, die Verschiebung sei von der Union ausgegangen. Die Arbeitsgruppe zu dem Thema habe gute Vorarbeit geleistet.

Which sentence is more similar to the reference sentence?

- Annotations are dictionaries of tuples
- Example:
  - Ref=0, Sentence 1 = 1, Sentence 2 = 2
  - User selects 1: (1,2) += 1 in Sentence 0
  - User selects 2: (2,1) += 1 in Sentence 0

The linear case

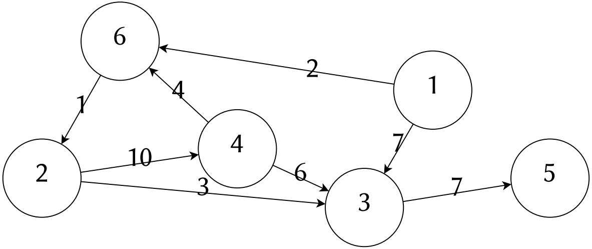


Expected result from a rational single user:

- Rank sentences, then divide by number of sentences ≈ score.
- Problems:
  - Users aren't rational
  - Users might disagree

What does a realistic result look like?

The realistic case



Problems:

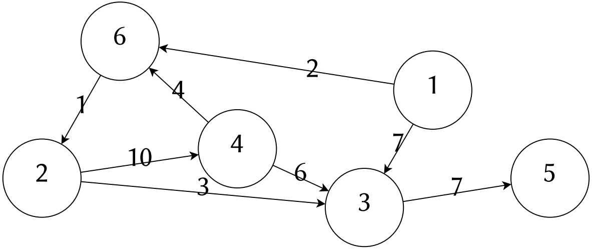
- Circular relationships exist
- Ranking is non obvious

Naive approach:

- Add up the edge weights for each node:
  - Node 3: 16, Node 5: 7
  - Does this match our expectations?

Better Solution?

The realistic case



Problems:

- Circular relationships exist
- Ranking is non obvious

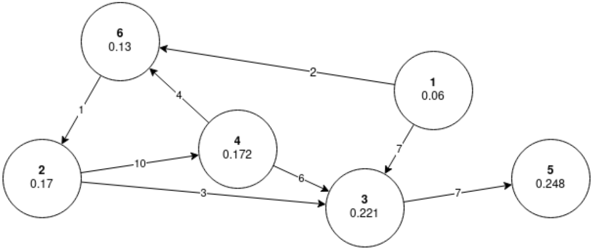
Naive approach:

- Add up the edge weights for each node:
  - Node 3: 16, Node 5: 7
  - Does this match our expectations?

Better Solution?

Page-Rank Algorithm[Pag+99]

The realistic case



Problems:

- Circular relationships exist
- Ranking is non obvious

Naive approach:

- Add up the edge weights for each node:
  - Node 3: 16, Node 5: 7
  - Does this match our expectations?

Better Solution?

Page-Rank Algorithm[Pag+99]

Suggested Page-Rank Implementation

- make sure reference sentence is rated to itself
- run page-rank over annotations
- norm the results over all sentences

Reference Sentence compared to itself $\longrightarrow$ upper bound.
Therefore this should give us decent scores.
Advantages:

- Easier to annotate than 1-5 scale.
- Quicker to annotate.
- Direct comparison data exists for sentence pairs.

Literatur I

[Dev+18] Jacob Devlin, Ming-Wei Chang, Kenton Lee und Kristina Toutanova. „Bert: Pre-training of deep bidirectional transformers for language understanding“. In: *arXiv preprint arXiv:1810.04805* (2018).

[Emm17] Chris Emmery. *Euclidean vs. Cosine Distance.* 25. März 2017. URL: `https://cmry.github.io/notes/euclidean-v-cosine` (besucht am 03. 02. 2020).

[Kus+15] Matt Kusner, Yu Sun, Nicholas Kolkin und Kilian Weinberger. „From word embeddings to document distances“. In: *International conference on machine learning*. 2015, S. 957–966.

[LM14] Quoc Le und Tomas Mikolov. „Distributed representations of sentences and documents“. In: *International conference on machine learning*. 2014, S. 1188–1196.

[Mik+15] Tomas Mikolov, Kai Chen, Gregory S Corrado und Jeffrey A Dean. *Computing numeric representations of words in a high-dimensional space.* US Patent 9,037,464. 2015.

Literatur II

[Pag+99] Lawrence Page, Sergey Brin, Rajeev Motwani und Terry Winograd. *The pagerank citation ranking: Bringing order to the web.* Techn. Ber. Stanford InfoLab, 1999.

Outline

- Kurzüberblick LaTeX-Beamer
    - Strukturierung
- Implementation
    - The Architecture
- Results
    - Results
    - Lessons Learned
- The Annotator
    - Interface
- The Annotator
    - Ranking
- The Annotator
    - Ranking
- Literatur