Document Similarity Benchmark

Jan-Gabriel Mylius

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

10. February 2020, 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.

The Architecture

The benchmark suite consists of two essential elements:

- the algorithms
- the datasets

We want a framework:

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

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]

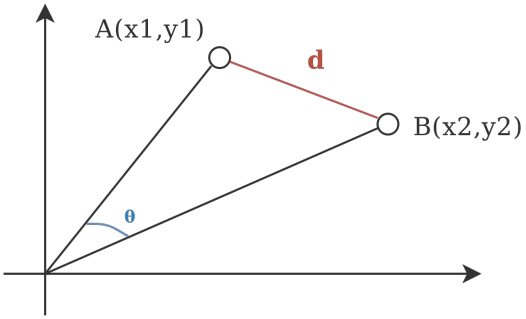
Functional Principle

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:

- Cosine similarity
- Euclidean similarity
- Jaccard similarity



Euclidean vs Cosine Distance
Source:
Chris Emmery [Emm17]

The dataset class

The dataset class has the following essential functions:

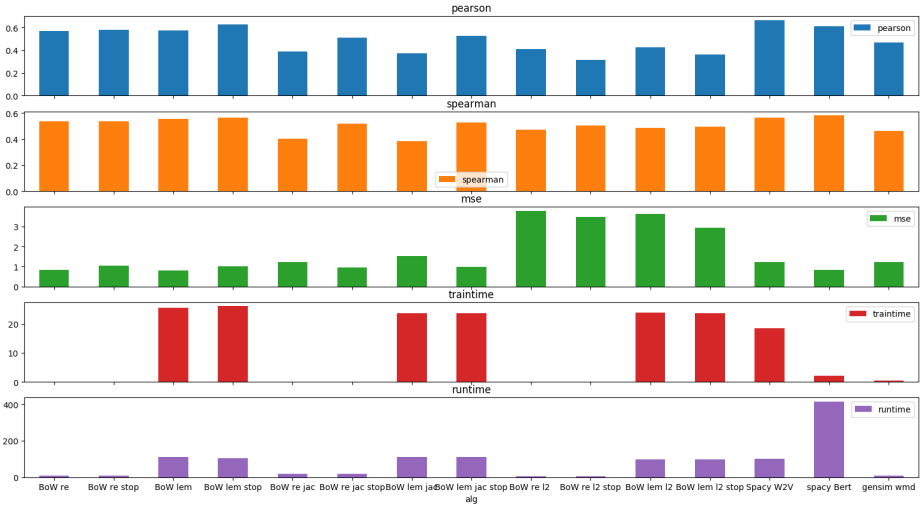
- loads a dataset
  - STS
  - SICK
  - custom
- runs an algorithm
- norms results to match the datasets scale
- stores results for all algorithms that have run

The benchmark

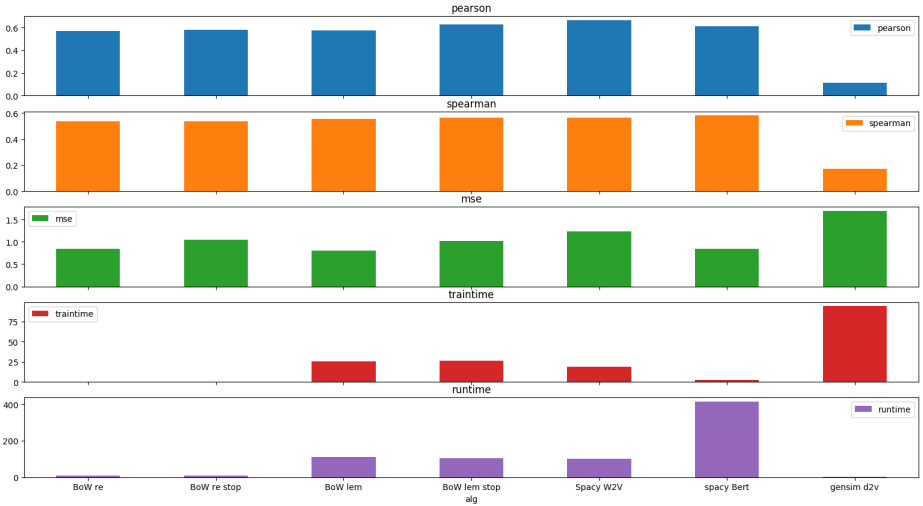
What the benchmark does:

- 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 as json file

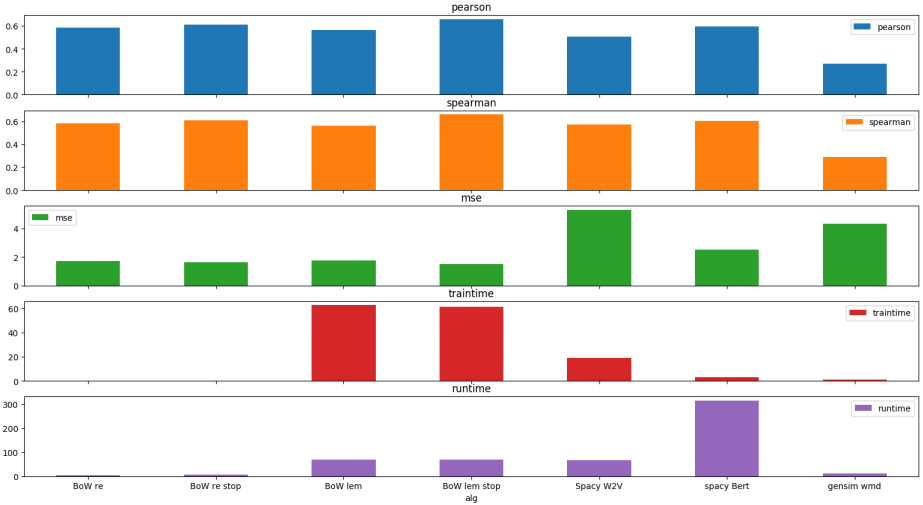
Overview of all results



Overview of sick results without l2 and jaccard



Overview of sts results without l2 and jaccard



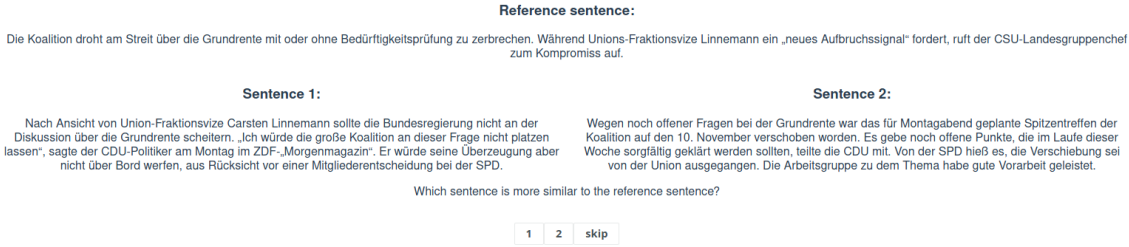
Discussion of Results

- l2 and jaccard similarity not useful
- Filtering stopwords improves results
- Lemmatisation improves results
- Doc2Vec needs more training data
  - Maybe allow additional training data in future version
  - There is no currently compatible pretrained Doc2Vec model
    - → Some one should train one for current gensim
- BERT and Word2Vec not reliably better than BoW with lemmatisation

Lessons Learned

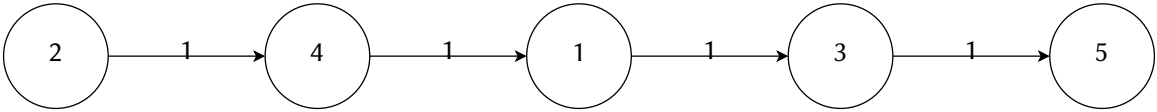
- More and larger datasets would be interesting to have
- Semantic similarity is hard to quantify
  - STS rating instructions don't match my intuition
    → were annotators consistent?
  - Finding scale people agree potentially difficult
    → annotators need a lot of training
  → different annotation method might be better

Interface



- Annotations are dictionaries of tuples
- Allows creation of different task format
- Annotations can be viewed as graph for each reference
  - Nodes: sentences
  - Edges: more similar relationship
  - Edge weights: number of annotators agreeing

The linear case

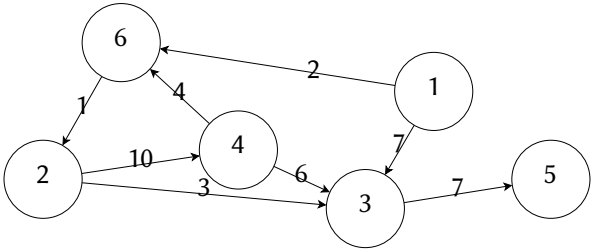


Expected result from a rational single user:

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

What does a realistic result look like?

The realistic case



Problems:

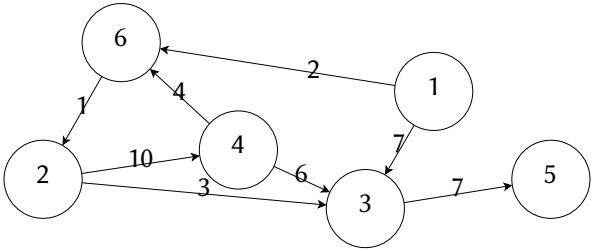
- Circular relationships exist
- Ranking is non obvious

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

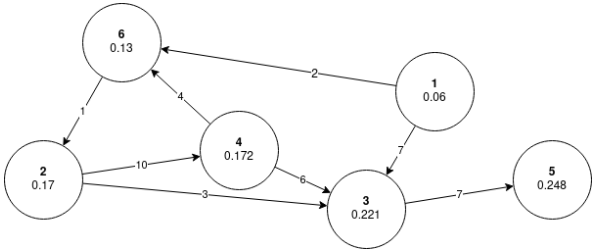
Naive approach:

- 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