Flexible Similarity Search of Semantic Vectors Using Fulltext Search Engines

Michal Růžička, Vít Novotný, Petr Sojka; Jan Pomikálek, Radim Řehůřek

Masaryk University, Faculty of Informatics, Brno, Czech Republic
mruzicka@mail.muni.cz, witiko@mail.muni.cz, sojka@fi.muni.cz;
RaRe Technologies
honza@rare-technologies.com, radim@rare-technologies.com

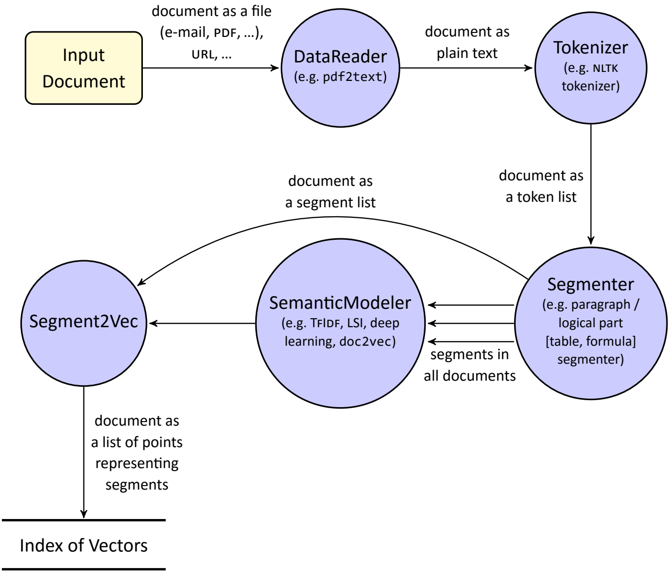
https://mir.fi.muni.cz/ https://rare-technologies.com/

Illustrations by Jiří Franek.

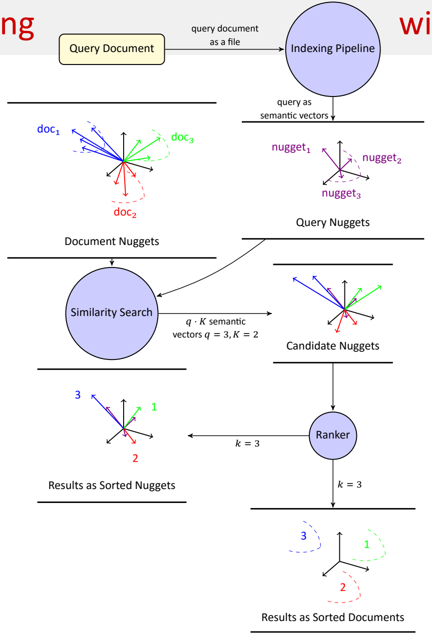
Outline

Outline

Semantic Indexing



Semantic Searching with Nuggets



Re-Ranking Techniques

1. **Fast:** find candidate nuggets via Elasticsearch.
2. **Slow but precise:** re-rank candidate nuggets with exact similarity metric.
    - Cosine similarity.
    - Euclidean similarity.

Re-Ranking Techniques

1. **Fast:** find candidate nuggets via Elasticsearch.
2. **Slow but precise:** re-rank candidate nuggets with exact similarity metric.
   - Cosine similarity.
   - Euclidean similarity.
   - ...

Outline

String Encoding of Semantic Vectors

- Encoding of semantic vectors to strings (feature tokens):
  - Semantic vector of three dimensions:

$$\vec{w} = [0.12, -0.13, 0.065]$$

String Encoding of Semantic Vectors

- Encoding of semantic vectors to strings (feature tokens):
  - Semantic vector of three dimensions:

    $$\vec{w} = [0.12, -0.13, 0.065]$$

  - Rounding to two decimal places, string encoded:

    $$\vec{w} = [0.12, -0.13, 0.065]$$

String Encoding of Semantic Vectors

- Encoding of semantic vectors to strings (feature tokens):
  - Semantic vector of three dimensions:

    $$\vec{w} = [0.12, -0.13, 0.065]$$

  - Rounding to two decimal places, string encoded:

    $$\vec{w} = [\text{'}\mathtt{0}\text{'}\ 0.12,\ -0.13,\ 0.065]$$

String Encoding of Semantic Vectors

- Encoding of semantic vectors to strings (feature tokens):
  - Semantic vector of three dimensions:

$$\vec{w} = [0.12, -0.13, 0.065]$$

  - Rounding to two decimal places, string encoded:

$$\vec{w} = [\text{'0'}\ 0.12,\ \text{'1'}\ -0.13,\ 0.065]$$

String Encoding of Semantic Vectors

- Encoding of semantic vectors to strings (feature tokens):
  - Semantic vector of three dimensions:

    $$\vec{w} = [0.12, -0.13, 0.065]$$

  - Rounding to two decimal places, string encoded:

    $$\vec{w} = [\text{'0'}\ 0.12, \text{'1'}\ -0.13, \text{'2'}\ 0.065]$$

String Encoding of Semantic Vectors

- Encoding of semantic vectors to strings (feature tokens):
  - Semantic vector of three dimensions:

    $$\vec{w} = [0.12, -0.13, 0.065]$$

  - Rounding to two decimal places, string encoded:

    $$\vec{w} = [\text{'0P2'}\ 0.12, \text{'1'}\ -0.13, \text{'2'}\ 0.065]$$

String Encoding of Semantic Vectors

- Encoding of semantic vectors to strings (feature tokens):
  - Semantic vector of three dimensions:

    $$\vec{w} = [0.12, -0.13, 0.065]$$

  - Rounding to two decimal places, string encoded:

    $$\vec{w} = [\texttt{'0P2'}\ 0.12, \texttt{'1P2'}\ -0.13, \texttt{'2'}\ 0.065]$$

String Encoding of Semantic Vectors

- Encoding of semantic vectors to strings (feature tokens):
  - Semantic vector of three dimensions:

    $$\vec{w} = [0.12, -0.13, 0.065]$$

  - Rounding to two decimal places, string encoded:

    $$\vec{w} = [\text{'0P2'}\ 0.12, \text{'1P2'}\ -0.13, \text{'2P2'}\ 0.07]$$

String Encoding of Semantic Vectors

- Encoding of semantic vectors to strings (feature tokens):
  - Semantic vector of three dimensions:

    $$\vec{w} = [0.12, -0.13, 0.065]$$

  - Rounding to two decimal places, string encoded:

    $$\vec{w} = [\texttt{'0P2i0d12'}, \texttt{'1P2'}\ -0.13, \texttt{'2P2'}\ 0.07]$$

String Encoding of Semantic Vectors

- Encoding of semantic vectors to strings (feature tokens):
  - Semantic vector of three dimensions:

    $$\vec{w} = [0.12, -0.13, 0.065]$$

  - Rounding to two decimal places, string encoded:

    $$\vec{w} = [\text{'0P2i0d12'}, \text{'1P2ineg0d13'}, \text{'2P2'}\ 0.07]$$

String Encoding of Semantic Vectors

- Encoding of semantic vectors to strings (feature tokens):
  - Semantic vector of three dimensions:

$$\vec{w} = [0.12, -0.13, 0.065]$$

  - Rounding to two decimal places, string encoded:

$$\vec{w} = [\texttt{'0P2i0d12'}, \texttt{'1P2ineg0d13'}, \texttt{'2P2i0d07'}]$$

String Encoding of Semantic Vectors

- Encoding of semantic vectors to strings (feature tokens):
  - Semantic vector of three dimensions:

    $$\vec{w} = [0.12, -0.13, 0.065]$$

  - Rounding to two decimal places, string encoded:

    $$\vec{w} = [\texttt{'0P2i0d12'}, \texttt{'1P2ineg0d13'}, \texttt{'2P2i0d07'}]$$

  - Feature tokens:
    - `0P2i0d12`
    - `1P2ineg0d13`
    - `2P2i0d07`

High-Pass Filtering – Speed Optimization

- High-pass filtering: $\vec{w} = [0.12, -0.13, 0.065]$

  trim Fixed threshold, for example 0.1:
  Keep only 0.12, -0.13 from $\vec{w}$, as $|0.065| < 0.1$.

  best Fixed number of the best values is used, for example only the best one:
  Keep only -0.13 from $\vec{w}$, as $|-0.13|$ is the highest absolute value in $\vec{w}$.

High-Pass Filtering – Speed Optimization

- High-pass filtering: $\vec{w} = [0.12, -0.13, 0.065]$

trim Fixed threshold, for example 0.1:
Keep only 0.12, -0.13 from $\vec{w}$, as $|0.065| < 0.1$.

best Fixed number of the best values is used, for example only the best one:
Keep only -0.13 from $\vec{w}$, as $|-0.13|$ is the highest absolute value in $\vec{w}$.

Speed optimization of the search for candidate nuggets *without significant impact on the quality*.

Outline

Datasets

en-wiki — The English Wikipedia dataset.

- **LSA** with 400 dimensions
- **doc2vec** with 400 dimensions.

wiki-2014+gigaword-5 — Pre-trained word vectors from Wikipedia and English Gigaword Fifth Edition.

- **GloVe** with 50, 100, 200, and 300 dimensions.

common-crawl — Pre-trained word vectors from the Common Crawl project.

- **GloVe** with 300 dimensions.

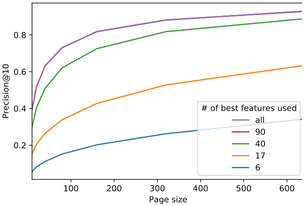
twitter — Pre-trained word vectors from the Twitter social network.

- **GloVe** with 25, 50, 100, and 200 dimensions.

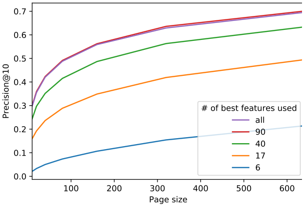
texmex — Image descriptors provided by the TEXMEX project.

- **SIFT** descriptors of images with 128 dimensions.

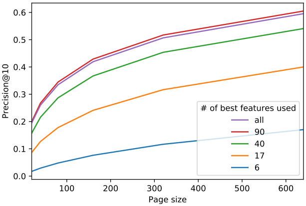
Comparison of Results



English Wikipedia
Cosine Similarity

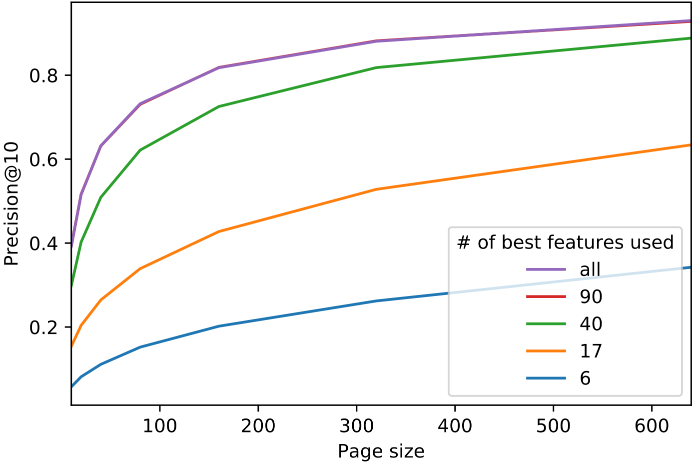


TEXMEX SIFT Descriptors
Cosine Similarity



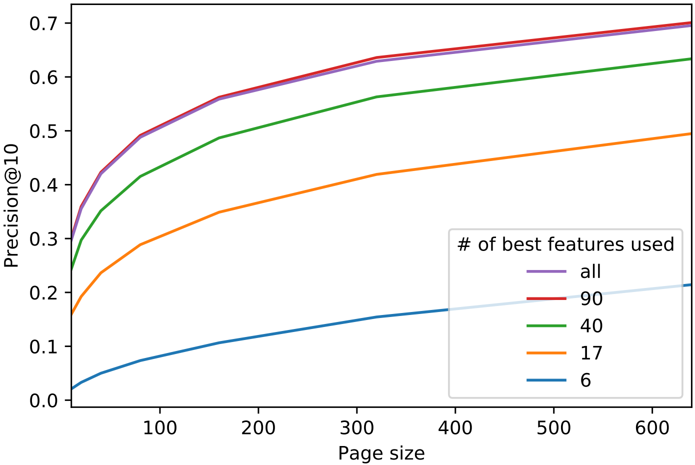
TEXMEX SIFT Descriptors
Euclidean Similarity

Comparison of Results



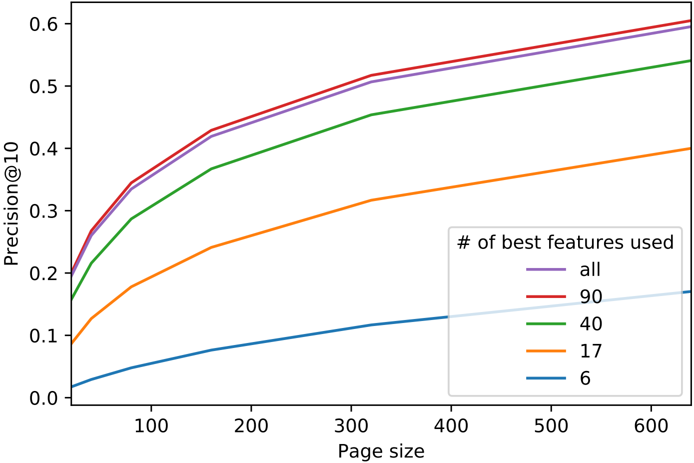
English Wikipedia Cosine Similarity

Comparison of Results



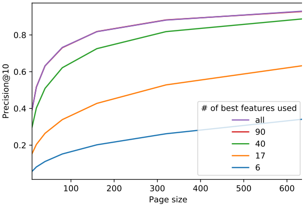
TEXMEX SIFT Descriptors Cosine Similarity

Comparison of Results

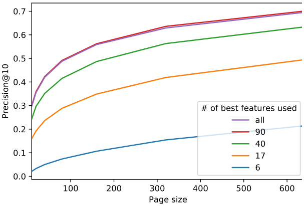


TEXMEX SIFT Descriptors Euclidean Similarity

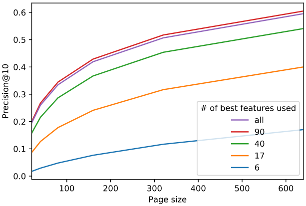
Comparison of Results



English Wikipedia
Cosine Similarity



TEXMEX SIFT Descriptors
Cosine Similarity

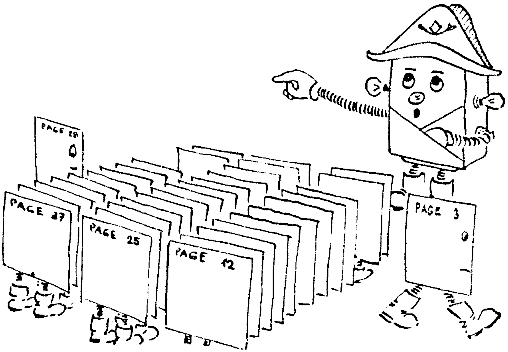


TEXMEX SIFT Descriptors
Euclidean Similarity

Summary

Flexible	Different input data formats / tokenizers / segmenters / semantic models / re-ranking methods / fulltext search engines / …
Similarity Search	Cosine / euclidean / … similarity.
of Semantic Vectors	LSI / deep learning / doc2vec / …
using Fulltext Search Engines	Sphinx, Lucene, Elasticsearch, Solr, …

Questions?



- Illustrations by Jiří Franek.
- RŮŽIČKA, Michal, Vít NOVOTNÝ, Petr SOJKA, Jan POMIKÁLEK and Radim ŘEHŮŘEK. Flexible Similarity Search of Semantic Vectors Using Fulltext Search Engines. In CEUR Workshop Proceedings, Vol. 1923. Vienna, Austria: Neuveden, 2017. p. 1–12, 12 pp. ISSN 1613-0073. `https://usc-isi-i2.github.io/ISWC17workshop/accepted-papers/HSSUES_2017_paper_2.pdf`
- RYGL, Jan, Jan POMIKÁLEK, Radim ŘEHŮŘEK, Michal RŮŽIČKA, Vít NOVOTNÝ and Petr SOJKA. Semantic Vector Encoding and Similarity Search Using Fulltext Search Engines. In Proceedings of the 2nd Workshop on Representation Learning for NLP. Vancouver, Canada: Association for Computational Linguistics, 2017. p. 81–90, 179 pp. ISBN 978-1-945626-62-3. DOI: `https://doi.org/10.18653/v1/W17-2611`
- RYGL, Jan, Petr SOJKA, Michal RŮŽIČKA and Radim ŘEHŮŘEK. ScaleText: The Design of a Scalable, Adaptable and User-Friendly Document System for Similarity Searches : Digging for Nuggets of Wisdom in Text. In Aleš Horák, Pavel Rychlý, Adam Rambousek. Proceedings of the Tenth Workshop on Recent Advances in Slavonic Natural Language Processing, RASLAN 2016. Brno: Tribun EU, 2016. p. 79–87, 9 pp. ISBN 978-80-263-1095-2. `https://nlp.fi.muni.cz/raslan/2016/paper08-Rygl_Sojka_etal.pdf`