ScaleText: The Design of a Scalable, Adaptable, and User-Friendly Document System for Similarity Searches
Digging for Nuggets of Wisdom in Text

Jan Rygl\textsuperscript{1}  Petr Sojka\textsuperscript{2}  Michal Růžička\textsuperscript{2}  Radim Řehůřek\textsuperscript{1}

\textsuperscript{1}RaRe Technologies, \{jimmy,radim\}@rare-technologies.com

\textsuperscript{2}Faculty of Informatics, Masaryk University, Brno, Czech Republic
sojka@fi.muni.cz and mruzicka@mail.muni.cz

December 2nd, 2016
Motivation
The Need for Truly *Semantic* Search

Indexing
Storing Document Chunks as Points in Vector Space

Similarity Search
Digging for Nuggets of Wisdom

Automatic Evaluation Framework for System Modules
  Bpref metrics

Conclusion and Future Work
Introduction

- Search as a gateway and primary access method for information in documents.
- From keyword based search to meaning based.
- From keyword based search to phrase/free question/paragraph based search.
- Topic modeling in large documents.
- Scalability—problem even with linear complexity.
Design Imperatives

- **Scalability**: with the size of today’s document collections, efficiency is a primary concern, allowing low latency responses.
- **Adaptability**: since no size fits all, the system should be easily customizable and tunable for any given application purpose.
- **Relevance**: search precision could be improved by clever semantic representations of the meanings of indexed texts. It is both necessary and desirable to find highly relevant document chunks.
- **Implementation Clarity**: the implementation should be written with ease of maintenance in mind.
- **Simplicity**: keep it simple stupid, yet provide the functionality needed.
Approaches

▪️ *a discrete representation* of meaning, which can be based on knowledge-based representations such as WordNet, BabelNet, Freebase or Wikipedia, or

▪️ *a smooth representation* in vector spaces based on a distributional hypothesis, e.g. representing meanings as word, phrase, sentence, ... embeddings (Mikolov, 2013) which are learned from the language used in big corpora by unsupervised, deep learning approaches, or by topic modeling (Blei, 2012)
Software Systems to support Semantic Similarity Search

- based on Gensim (Rehurek, Sojka, 2010)
- Kvasir
- Similarity search based on trees (M-tree) et al.
Indexing I
Storing Document Chunks as Points in Vector Space

ScaleText introduces a flexible data processing pipeline for document indexing, leading to semantic document representations in a vector space. The overall scheme of document transformations in the indexing workflow is depicted in Figure 1.

Figure 1: Data flow diagram of document indexing in ScaleText
Indexing II
Storing Document Chunks as Points in Vector Space

Segmenter (e.g. paragraph/logical part/table, formula segmenter) → SemanticModeler (e.g. TfIdf, LSI, deep learning, doc2vec) → Segment2Vec → Index of Vectors

document as a token list → document as a segment list

document as a segment list → all segments in all documents → document as a list of points representing segments

Figure 1: Data flow diagram of document indexing in ScaleText
The indexed dataset is used for similarity searching. To pursue the gold mining metaphor, gold nuggets are washed with different gold mining techniques. The overall schema of the search procedure is depicted in Figure 2.
Figure 2: Data flow diagram of document similarity search in ScaleText. $q$ is the number of query nuggets, $K$ is the number of best nugget candidates for each query nugget, and $k$ is the number of desired results.
Figure 2: Data flow diagram of document similarity search in ScaleText. $q$ is the number of query nuggets, $K$ is the number of best nugget candidates for each query nugget, and $k$ is the number of desired results.
Figure 2: Data flow diagram of document similarity search in ScaleText. 
$q$ is the number of query nuggets, $K$ is the number of best nugget candidates for each query nugget, and $k$ is the number of desired results.
We implemented Bpref@$k$ as follows:

\[
B\text{pref}@k = \frac{1}{\min(R, k)} \sum_r \left( 1 - \frac{\min(\text{number of } n \text{ ranked higher than } r, R)}{\min(N, R)} \right),
\]

where

- $R$ is the number of documents relevant to the topic,
- $N$ is the number of documents irrelevant to the topic,
- $k$ is the maximal number of inspected results, and
- “number of $n$ ranked higher than $r$” is the number of irrelevant documents (according to the judgment) ranked higher than the relevant (according to the judgment) document $r$ that is being processed in the step.
Table 1: ScaleText prototype evaluation on the Enron dataset via Bpref. The single metric value is the average of Bpref@100 over all the queries.

<table>
<thead>
<tr>
<th>doc. model</th>
<th>document ranking strategy</th>
<th>#feat.</th>
<th>avg Bpref@100</th>
</tr>
</thead>
<tbody>
<tr>
<td>TfIdf</td>
<td>maximum nugget score</td>
<td>100</td>
<td>0.0451</td>
</tr>
<tr>
<td>TfIdf+LSI</td>
<td>maximum nugget score</td>
<td>50</td>
<td>0.0460</td>
</tr>
<tr>
<td>TfIdf+LSI</td>
<td>maximum nugget score</td>
<td>100</td>
<td>0.0565</td>
</tr>
<tr>
<td>TfIdf+LSI</td>
<td>maximum nugget score</td>
<td>500</td>
<td>0.0358</td>
</tr>
<tr>
<td>TfIdf</td>
<td>average nugget score</td>
<td>100</td>
<td>0.0451</td>
</tr>
<tr>
<td>TfIdf+LSI</td>
<td>average nugget score</td>
<td>50</td>
<td>0.0460</td>
</tr>
<tr>
<td>TfIdf+LSI</td>
<td>average nugget score</td>
<td>100</td>
<td>0.0548</td>
</tr>
<tr>
<td>TfIdf+LSI</td>
<td>average nugget score</td>
<td>500</td>
<td>0.0358</td>
</tr>
<tr>
<td>TfIdf</td>
<td>normalized sum of nugget scores</td>
<td>100</td>
<td>0.0451</td>
</tr>
<tr>
<td>TfIdf+LSI</td>
<td>normalized sum of nugget scores</td>
<td>50</td>
<td>0.0460</td>
</tr>
<tr>
<td>TfIdf+LSI</td>
<td>normalized sum of nugget scores</td>
<td>100</td>
<td>0.0534</td>
</tr>
<tr>
<td>TfIdf+LSI</td>
<td>normalized sum of nugget scores</td>
<td>500</td>
<td>0.0358</td>
</tr>
</tbody>
</table>
We have several research questions in our sights:

▶ **Word disambiguation in context:** current methods represent a word in the vector space as the centroid of its different meanings. We want to evaluate an approach based on random walks through texts so as to distinguish the representation of words in context.
We have several research questions in our sights:

- **Word disambiguation in context**: current methods represent a word in the vector space as the centroid of its different meanings. We want to evaluate an approach based on random walks through texts so as to distinguish the representation of words in context.

- **Compositionality of segment representation**: semantic vectors representing the meaning of segments should reflect compositionality of meaning of its parts, e.g. words, phrases and sentences.
We have several research questions in our sights:

- **Word disambiguation in context:** current methods represent a word in the vector space as the centroid of its different meanings. We want to evaluate an approach based on random walks through texts so as to distinguish the representation of words in context.

- **Compositionality of segment representation:** semantic vectors representing the meaning of segments should reflect compositionality of meaning of its parts, e.g. words, phrases and sentences.

- **Representation of narrativity:** we may represent narrative text qualities [5] as a trajectory of words or nuggets in vector space, e.g. document representation may be a trajectory instead of a point.
http://doi.acm.org/10.1145/2133806.2133826

http://doi.acm.org/10.1145/1008992.1009000


