Random Manhattan Indexing

Behrang Q. Zadeh and Siegfried Handschuh

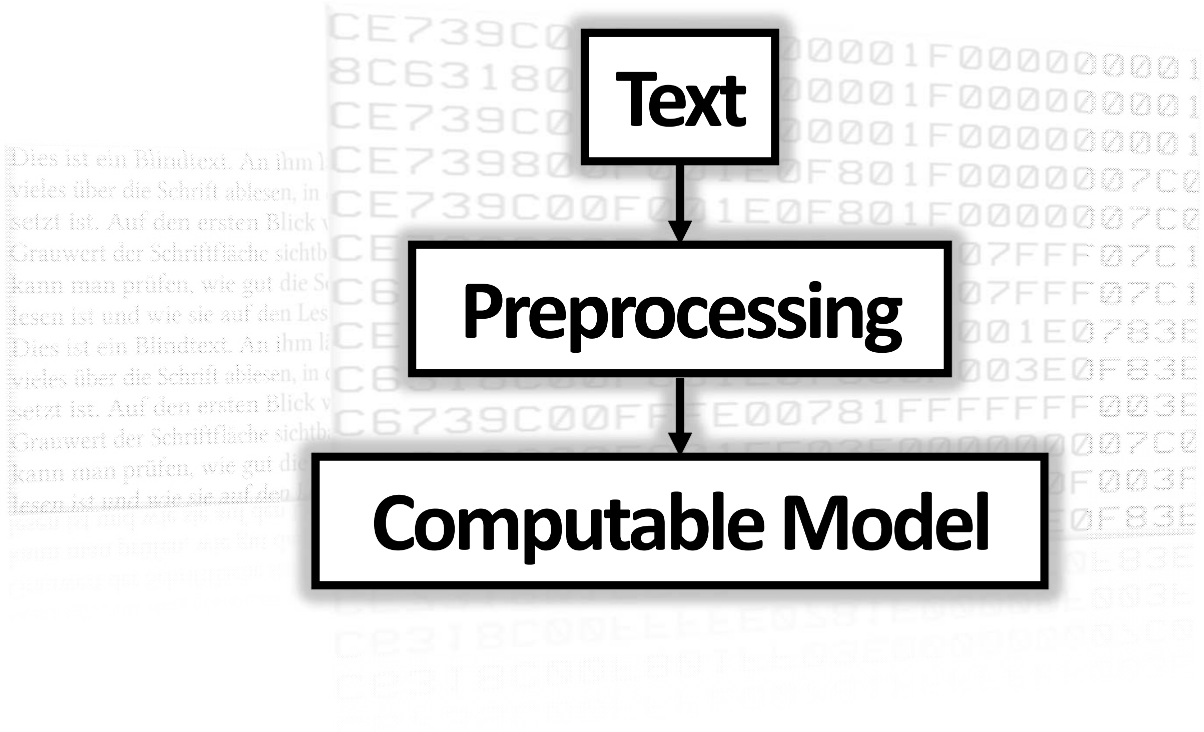
National University of Ireland, Galway, Ireland
University of Passau, Lower Bavaria, Germany

Processing Natural Language Text

- For Computers, natural language text is simply a sequence of bits and bytes.

Processing Natural Language Text

- For Computers, natural language text is simply a sequence of bits and bytes.



Vector Space Model (VSM)

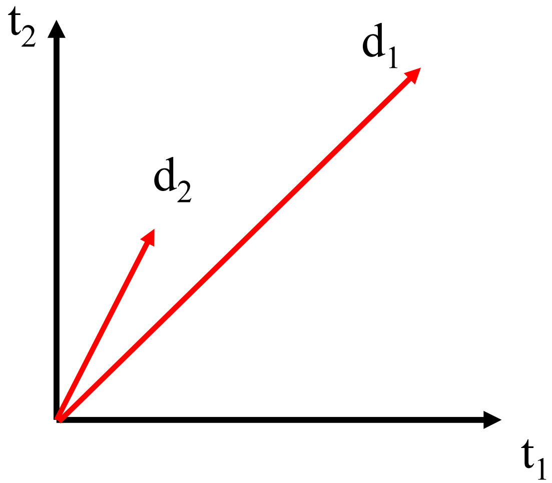
- Vector spaces are one of the models employed for text processing.
  - A Mathematical Structure $\langle V, \mathbb{R}, +, \cdot \rangle$ that satisfy certain axioms.
- Text elements are converted to real-valued vectors.
- Each dimension of the vector represents some information about text elements.

VSMs: an Example

- In a text-based information retrieval task:
  - Each dimension of the VSM represents an index term *t*.
  - Documents are represented by vectors.

Salton et. al. (1975)

VSMs: an Example



$$\longleftrightarrow \quad \begin{array}{c c} & \begin{array}{cc} t_1 & t_2 \end{array} \\ \begin{array}{c} d_1 \\ d_2 \end{array} & \begin{pmatrix} 4 & 4 \\ 1 & 2 \end{pmatrix} \end{array}$$

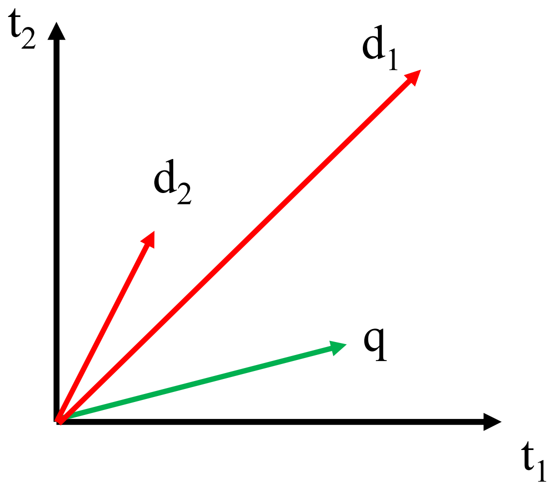
Salton et. al. (1975)

VSMs: an Example

- In a text-based information retrieval task:
    - Each dimension of the VSM represents an index term *t*.
    - Documents are represented by vectors.
    - Queries are treated as pseudo-documents.

Salton et. al. (1975)

VSMs: an Example



$$
\longleftrightarrow
\quad
\begin{array}{c}
\\ d_1 \\ d_2 \\ q
\end{array}
\begin{array}{cc}
t_1 & t_2 \\
\end{array}
$$

$$
\begin{array}{c|cc}
 & t_1 & t_2 \\
\hline
d_1 & 4 & 4 \\
d_2 & 1 & 2 \\
q & 3 & 1
\end{array}
$$

Salton et. al. (1975)

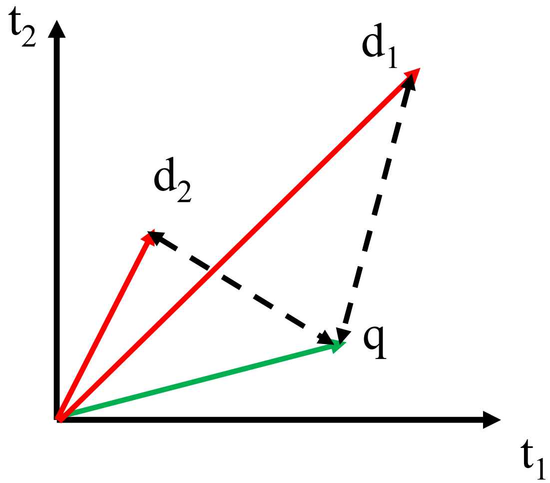
VSMs: an Example

- In a text-based information retrieval task:
  - Each dimension of the VSM represents an index term *t*.
  - Documents are represented by vectors.
  - Queries are treated as pseudo-documents.
  - Using a norm structure, a notion of distance is defined and used to assess the similarity between vectors, i.e. documents and queries.

Salton et. al. (1975)

VSMs: an Example

The L2 or Euclidean norm, $\|v\|_2 = \sqrt{\sum_i v_{\mathrm{i}}^{\,2}}$



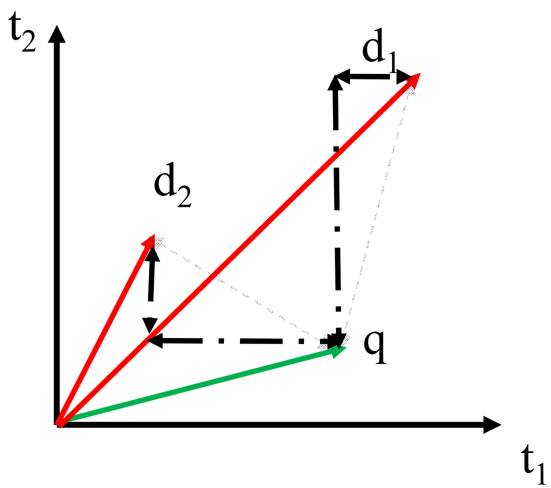
$$\begin{array}{cc} & \begin{array}{cc} t_1 & t_2 \end{array} \\ \begin{array}{c} d_1 \\ d_2 \\ q \end{array} & \begin{pmatrix} 4 & 4 \\ 1 & 2 \\ 3 & 1 \end{pmatrix} \end{array}$$

$$dist_2(d_1, q) = \|d_1 - q\|_2 = \sqrt{(4-3)^2 + (4-1)^2} = \sqrt{10}$$
$$dist_2(d_2, q) = \|d_2 - q\|_2 = \sqrt{(1-3)^2 + (2-1)^2} = \sqrt{5}$$

Salton et. al. (1975)

VSMs: an Example

The L1 norm, $\|v\|_1 = \sum_i |v_i|$



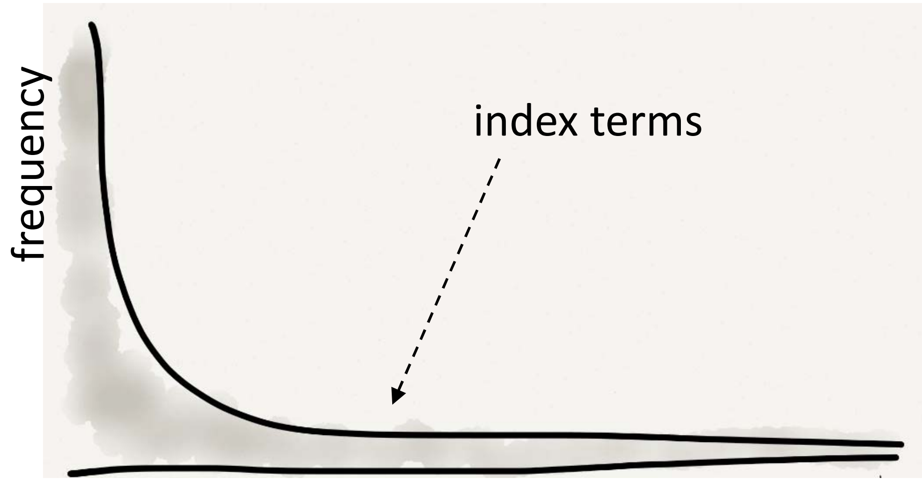
$$\begin{array}{c} \\ d_1 \\ d_2 \\ q \end{array} \begin{array}{c} \begin{array}{cc} t_1 & t_2 \end{array} \\ \begin{pmatrix} 4 & 4 \\ 1 & 2 \\ \boxed{3 \quad 1} \end{pmatrix} \end{array}$$

$$dist_1(d_1, q) = \|d_1 - q\|_1 = |(4-3)| + |(4-1)| = 4$$

$$dist_1(d_2, q) = \|d_2 - q\|_1 = |(1-3)| + |(2-1)| = 3$$

The dimensionality barrier

- Due to the Zipfian distribution of index terms, the number of index terms escalates when new documents are added.
- Adding new index terms requires adding additional dimensions to the vector space.

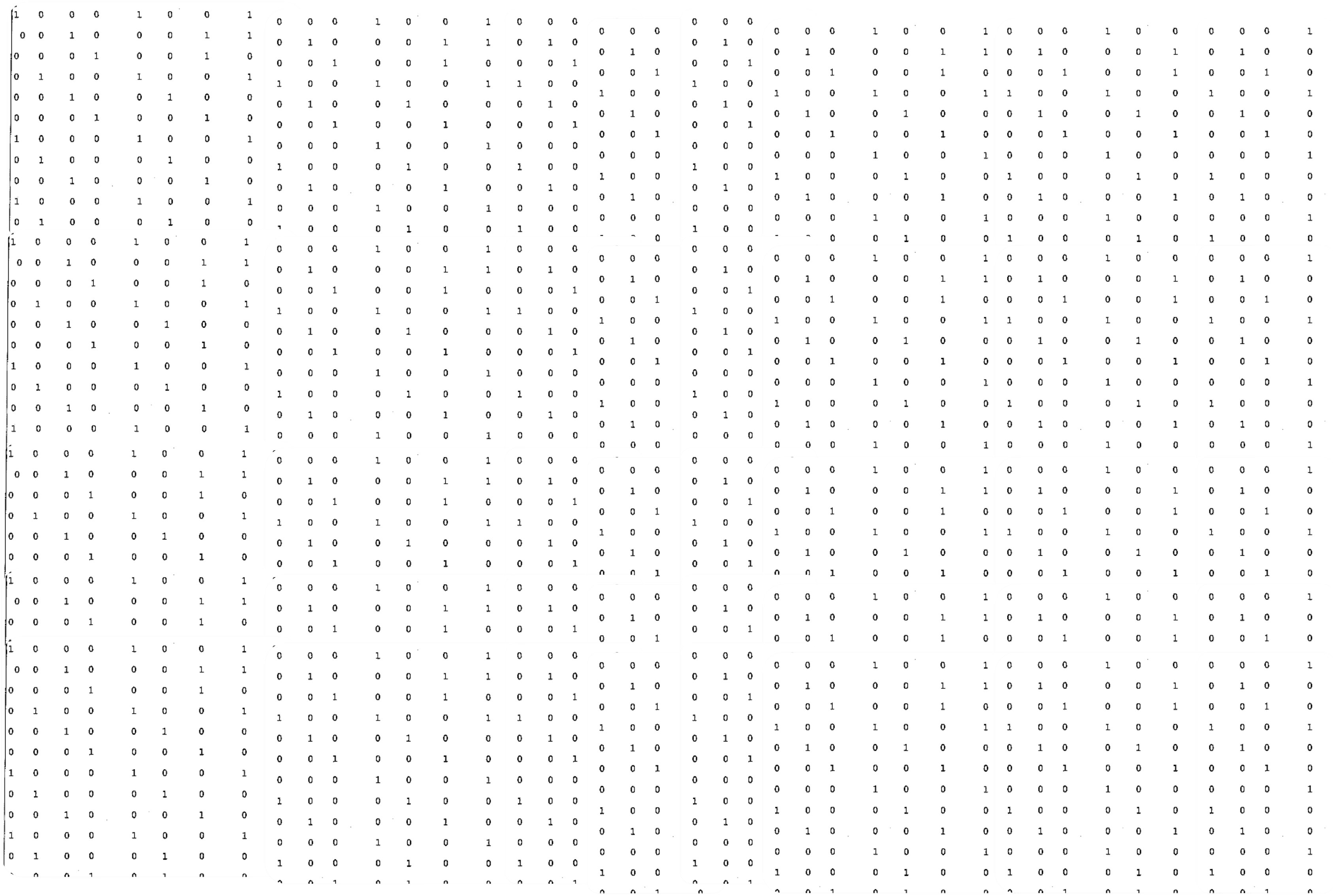


The dimensionality barrier

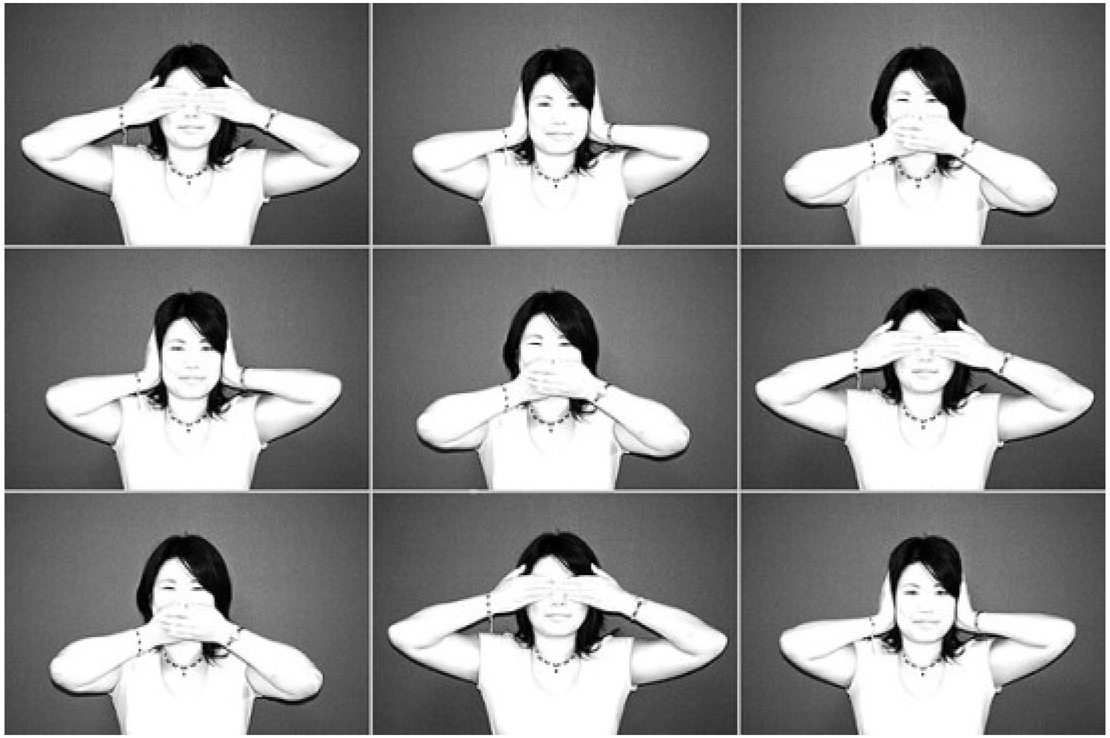
- Due to the Zipfian distribution of index terms, the number of index terms escalates when new documents are added.
- Adding new index terms requires adding additional dimensions to the vector space.
- Therefore, VSMs are extremely high-dimensional and sparse:

“THE CURSE OF DIMENSIONALITY”

The dimensionality barrier



Overcoming the dimensionality barrier



Overcoming the dimensionality barrier

- Dimensionality Reduction Techniques are employed to resolve the problem.

Overcoming the dimensionality barrier

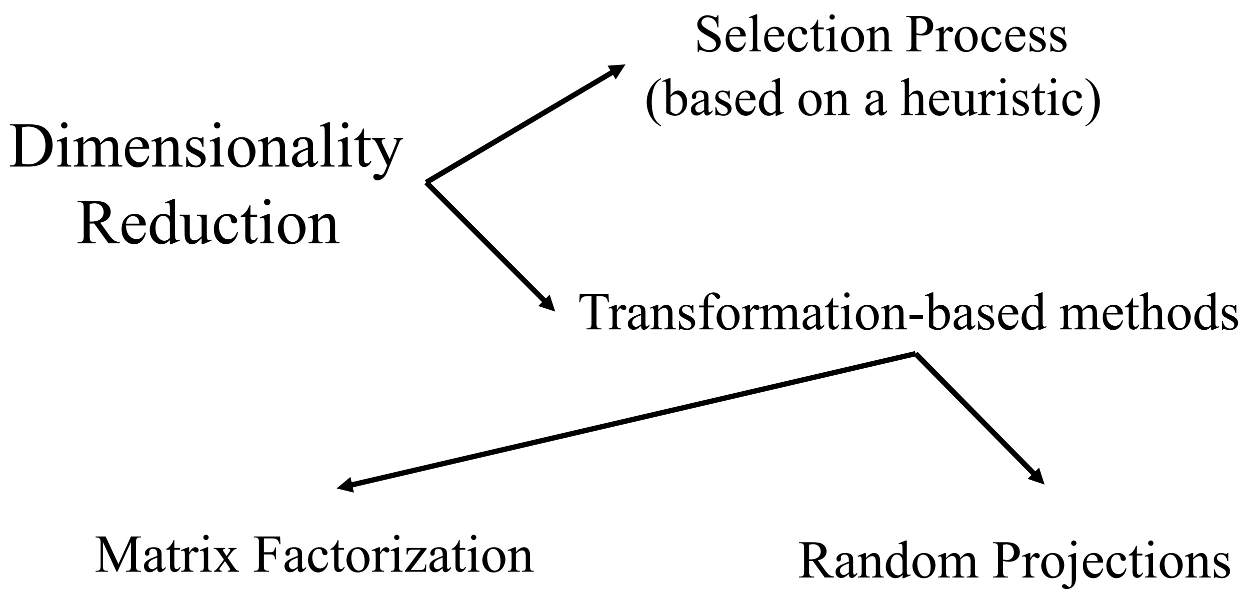
- Dimensionality Reduction Techniques are employed to resolve the problem.

Dimensionality Reduction → Selection Process (based on a heuristic)

Dimensionality Reduction → Transformation-based methods

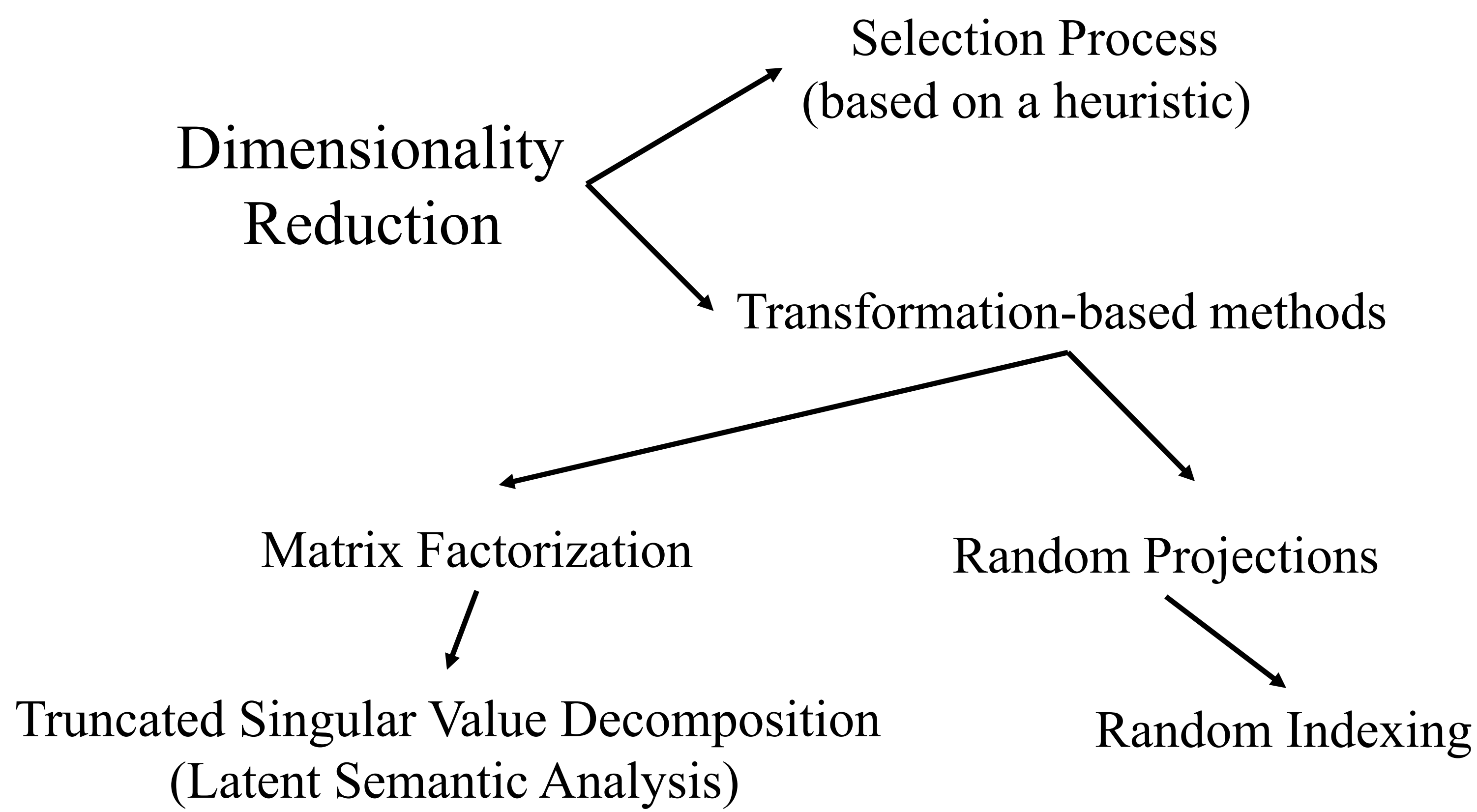
Overcoming the dimensionality barrier

- Dimensionality Reduction Techniques are employed to resolve the problem.



Overcoming the dimensionality barrier

- Dimensionality Reduction Techniques are employed to resolve the problem.



Limitations of Matrix Factorization

- Computation of Singular Value Decomposition (SVD) is process-intensive, i.e. $O(mn^2)$.
- Every time a VSM is updated, SVD must be recomputed:
  - Not suitable for big-text data analytics.
  - Not suitable for frequently updated text-data, e.g. blogs, tweeter analysis, etc.

Limitations of Matrix Factorization

- Computation of Singular Value Decomposition (SVD) is process-intensive, i.e. $O(mn^2)$.
- Every time a VSM is updated, SVD must be recomputed:
  - Not suitable for big-text data analytics.
  - Not suitable for frequently updated text-data, e.g. blogs, tweeter analysis, etc.

Solution: Random projection methods skip the computation of transformations (eigenvectors).

Random Projections: Random Indexing

- VSM Dimensionality is decided independent of the number of index terms, i.e. dimensionality of a VSM is fixed.

- Algorithm:
  - Assign each index term to a randomly generated "index vector".
    - Most elements of index vectors are 0 and only a few +1 and -1, e.g. $r^{t1} = (0, 0, 1, 0, -1)$, $r^{t2} = (1, 0, -1, 0, 0)$, etc.
  - Assign each document to an empty "context vector".
  - Construct the VSM incrementally by the accumulation of index vectors to context vectors.

Random Indexing (Example)

Munich is the capital and largest city of the German state of Bavaria.

Document 1

Random Indexing (Example)

Munich is the capital and largest city of the German state of Bavaria.

Document 1

Munich
Capital
German
Bavaria
Largest
City
State

Index Terms

Random Indexing (Example)

Munich is the capital and largest city of the German state of Bavaria.

Document 1

D1 = (0, 0, 0, 0, 0)

Munich: (1,-1,0,0,0)
Capital: (0,0,1,0,-1)
Germany: (1,0,0,-1,0)
Bavaria: (0,0,0,1,-1)
Largest: (0,1,0,-1,0)
City: (1,0,-1,0,0)
State: (1,0,0,0,-1)

Index Terms

Random Indexing (Example)

Munich is the capital and largest city of the German state of Bavaria.

Document 1

D1 = (4, 0, 0, -1, -3)

Munich: (1,-1,0,0,0)
Capital: (0,0,1,0,-1)
Germany: (1,0,0,-1,0)
Bavaria: (0,0,0,1,-1)
Largest: (0,1,0,-1,0)
City: (1,0,-1,0,0)
State: (1,0,0,0,-1)

Index Terms

Munich	(	1	,-1	,0	,0	,0	)	+
Capital	(	0	,0	,1	,0	,-1	)	+
Germany	(	1	,0	,0	,-1	,0	)	+
Bavaria	(	0	,0	,0	,1	,-1	)	+
Largest	(	0	,1	,0	,-1	,0	)	+
City	(	1	,0	,-1	,0	,0	)	+
State	(	1	,0	,0	,0	,-1	)	
D1 =	**(**	**4**	**,0**	**, 0**	**, -1**	**, -3**	**)**	

Random Indexing (Example)

Munich is the capital and largest city of the German state of Bavaria.

Document 1

D1 = (4, 0, 0, -1, -3)

Munich is a beautiful, historical city.

Document 2

Munich: (1,-1,0,0,0)
Capital: (0,0,1,0,-1)
Germany: (1,0,0,-1,0)
Bavaria: (0,0,0,1,-1)
Largest: (0,1,0,-1,0)
City: (1,0,-1,0,0)
State: (1,0,0,0,-1)

Index Terms

Random Indexing (Example)

Munich is the capital and largest city of the German state of Bavaria.

Document 1

D1 = (4, 0, 0, -1, -3)

Munich is a beautiful, historical city.

Document 2

D2 = (0, 0, 0, 0, 0)

Munich: (1,-1,0,0,0)
Capital: (0,0,1,0,-1)
Germany: (1,0,0,-1,0)
Bavaria: (0,0,0,1,-1)
Largest: (0,1,0,-1,0)
City: (1,0,-1,0,0)
State: (1,0,0,0,-1)
Beautiful: (0,0,1,-1,0)
Historical: (0,1,0,0,-1)

Index Terms

Random Indexing (Example)

Munich is the capital and largest city of the German state of Bavaria.

Document 1

D1 = (4, 0, 0, -1, -3)

Munich is a beautiful, historical city.

Document 2

D2 = (2,0,0, -1,-1)

Munich: (1,-1,0,0,0)
Capital: (0,0,1,0,-1)
Germany: (1,0,0,-1,0)
Bavaria: (0,0,0,1,-1)
Largest: (0,1,0,-1,0)
City: (1,0,-1,0,0)
State: (1,0,0,0,-1)
Beautiful: (0,0,1,-1,0)
Historical: (0,1,0,0,-1)

Index Terms

Random Indexing (Example)

Munich is the capital and largest city of the German state of Bavaria.

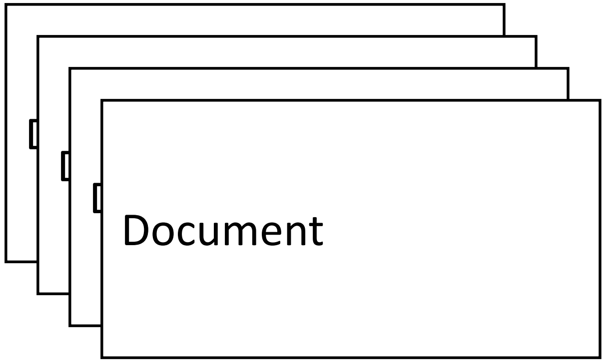
Document 1

D1 = (4, 0, 0, -1, -3)

Munich is a beautiful, historical city.

Document 2

D2 = (2,0,0, -1,-1)



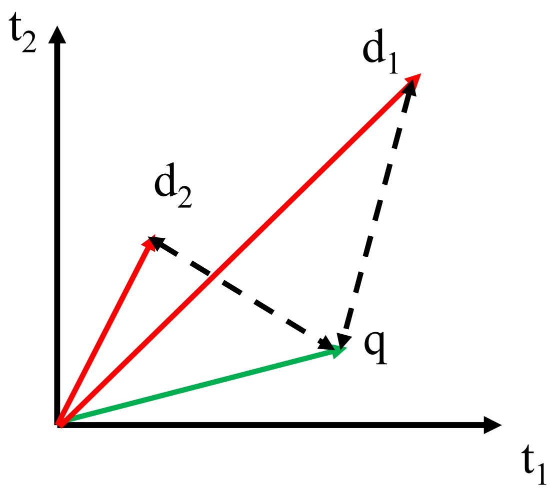
Munich: (1,-1,0,0,0)
Capital: (0,0,1,0,-1)
Germany: (1,0,0,-1,0)
Bavaria: (0,0,0,1,-1)
Largest: (0,1,0,-1,0)
City: (1,0,-1,0,0)
State: (1,0,0,0,-1)
Beautiful: (0,0,1,-1,0)
Historical: (0,1,0,0,-1)
.
.
.
.

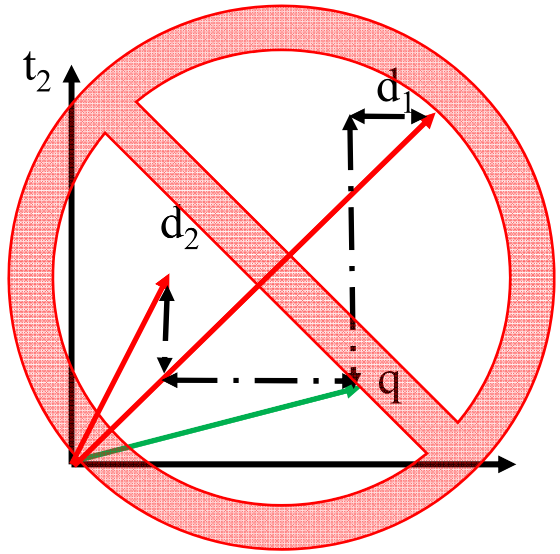
Index Terms

FIXED DIMENSION

Limitation of Random Indexing

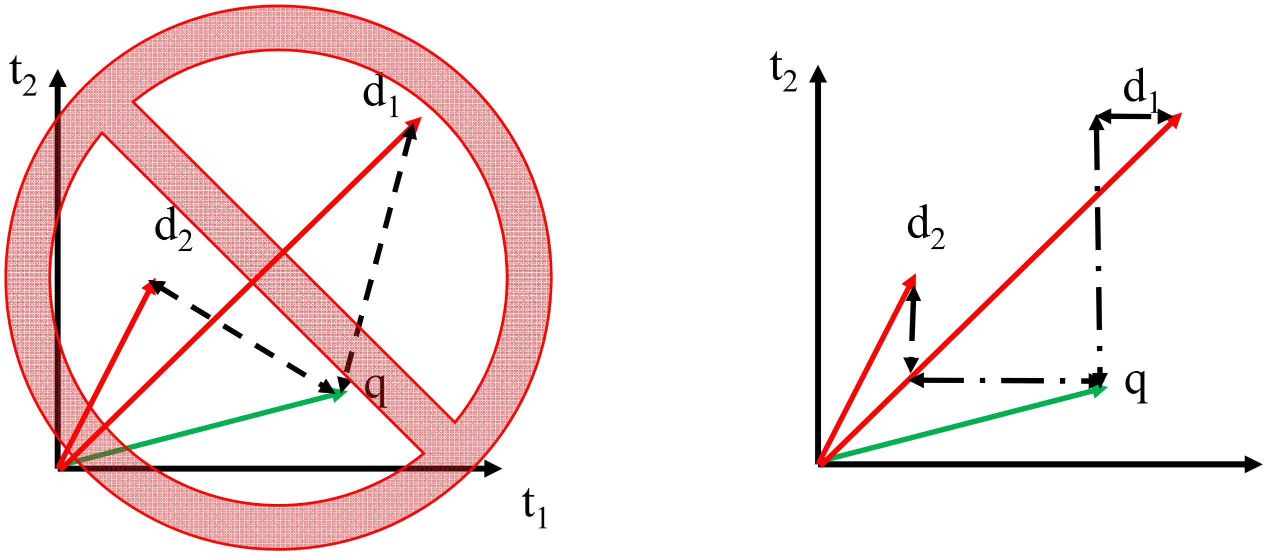
- Random Indexing employs a Gaussian Random Projection.
- A Gaussian Random Projection can only be used for the estimation of Euclidean distances.





Random Manhattan Indexing

- In an applications we may want to estimate the L1 (Manhattan) distance between vectors.
- Random Manhattan Indexing (RMI) can be used to estimate the Manhattan distances.



The RMI method

- RMI employs Cauchy Random Projections
- RMI is an incremental technique (similar to RI):
  - Each index term is assigned to an index vectors.
  - Index vectors are high-dimensional and randomly generated with the following distribution:

$$r_{ij} = \begin{cases} \frac{1}{U_1} & \text{With probability } \frac{s}{2} \\ 0 & \text{With probability } 1 - s \\ -\frac{1}{U_2} & \text{With probability } \frac{s}{2} \end{cases}$$

U_1 and U_2 are independent uniform random variables in (0, 1).

Indyk (2000), Li (2008)

The RMI method

- Assign documents to context vectors.
- Create context vectors incrementally.
- Estimate the Manhattan distance between vectors using the following (non-linear) equation:

$$\widehat{dist}_1(u,v) = \exp\left(\frac{1}{m}\sum_{i=1}^{m} \ln(|u_i - v_i|)\right)$$

m is the dimension of the RMI-constructed VSM

RMI (Example)

Munich is the capital and largest city of the German state of Bavaria.

Document 1

D1 = (1.3,5.89,-6.7,-3.5,-8)

Munich: (0,0,1.0,-2.1,0)
Capital: (0,1.49,0,-1.2,0)
German: (0,0,0,1.9,-1.8)
Bavaria: (1.3,0,0,0,-3.9)
Largest: (0,0,1.6,-2.1,0)
City: (0,2.5,0,0,-2.3)
State: (0,1.9,-9.3,0,0

Index Terms

RMI (Example)

Munich is the capital and largest city of the German state of Bavaria.

Document 1

D1 = (1.3,5.89,-6.7,-3.5,-8)

Munich: (0,0,1.0,-2.1,0)
Capital: (0,1.49,0,-1.2,0)
German: (0,0,0,1.9,-1.8)
Bavaria: (1.3,0,0,0,-3.9)
Largest: (0,0,1.6,-2.1,0)
City: (0,2.5,0,0,-2.3)
State: (0,1.9,-9.3,0,0

Index Terms

Use the non-linear estimator to assess similarities:

$$\widehat{dist}_1(u,v) = \exp\left(\frac{1}{m}\sum_{i=1}^{m} \ln(|u_i - v_i|)\right)$$

RMI's parameters

- The dimension of the RMI-constructed VSM.
- Number of non-zero elements in index vectors.

RMI's parameters

- The dimension of the RMI-constructed VSM:
  - It is independent of the number of index terms.
  - It is decided by the probability and the maximum expected distortion in distances.
  - For fixed set of documents, larger dimension results in less distortion.
  - According to our experiment, m=400 is suitable for most applications.

RMI's parameters

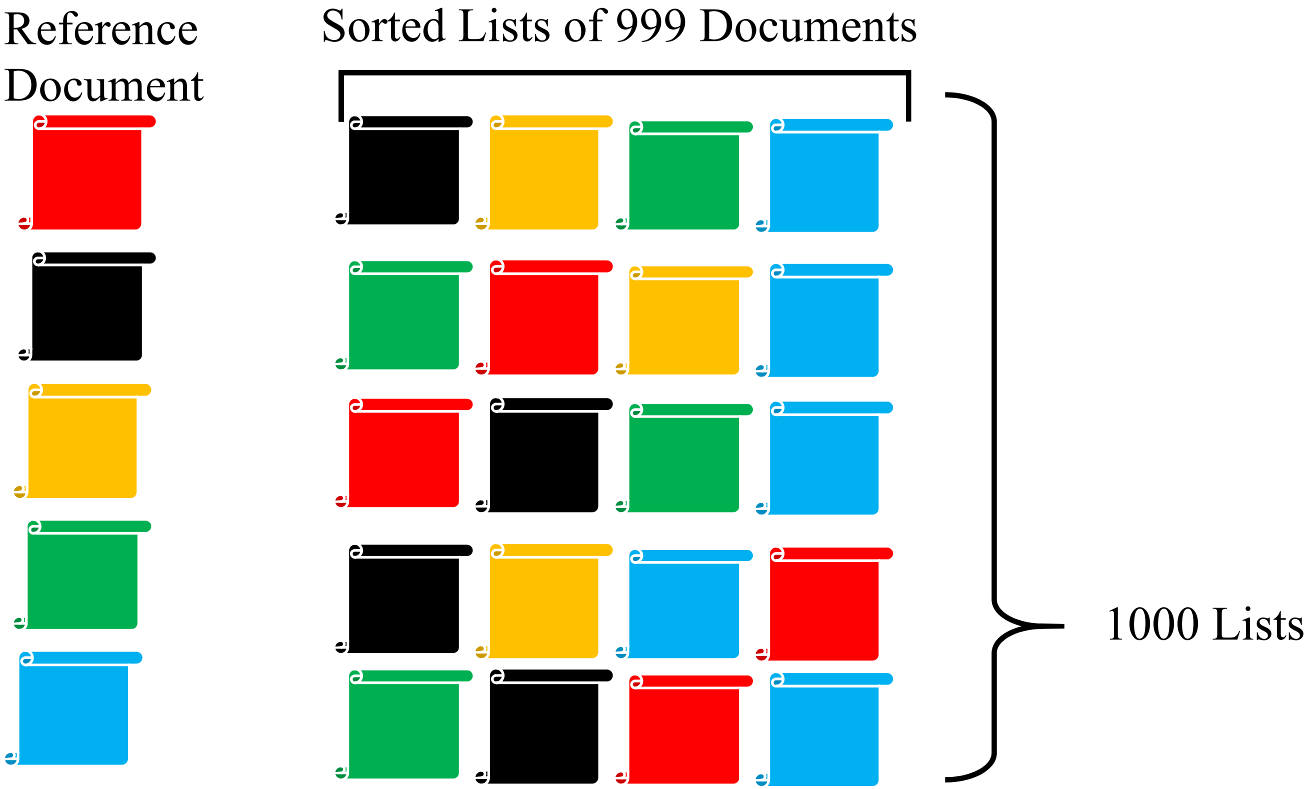
- The dimension of the RMI-constructed VSM.
- Number of non-zero elements in index vectors:
  - It is decided by the number of index terms and the sparsity of VSM at its original high-dimension.
  - We suggest $s = \frac{1}{O(\sqrt{\beta n})}$, where β is the sparseness of original high-dimensional VSM and n is the number index terms.
  - β is often considered to be around 0.0001 to 0.01.

Experimental Results

- We designed an experiment that shows the ability of RMI in preserving L1 distances:
  - A VSM is first constructed from the INEX-Wikipedia 2009 collection at its original high dimension (dimensionality of 2.53 million).
  - We choose a random list of 1000 random articles from the corpus.
  - The L1 distance of each document to the rest of documents in the list are calculated.
  - Documents are sorted by the calculated L1 distances.
  - The result is 1000 lists of 999 sorted documents.

Experimental Results

The set of 1000 random documents

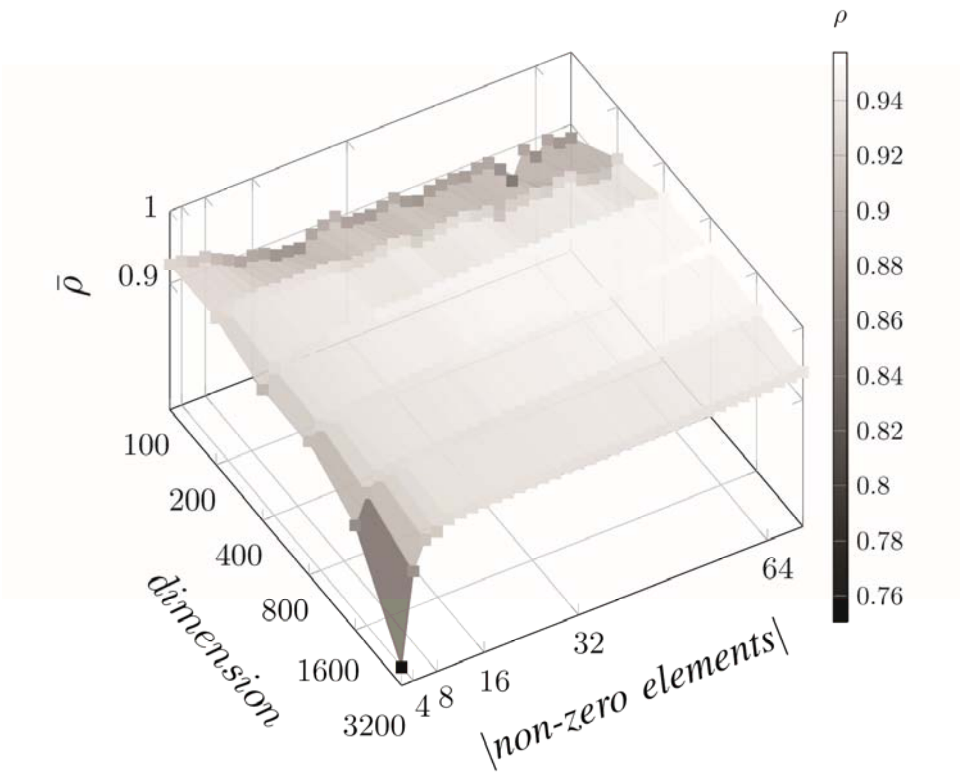


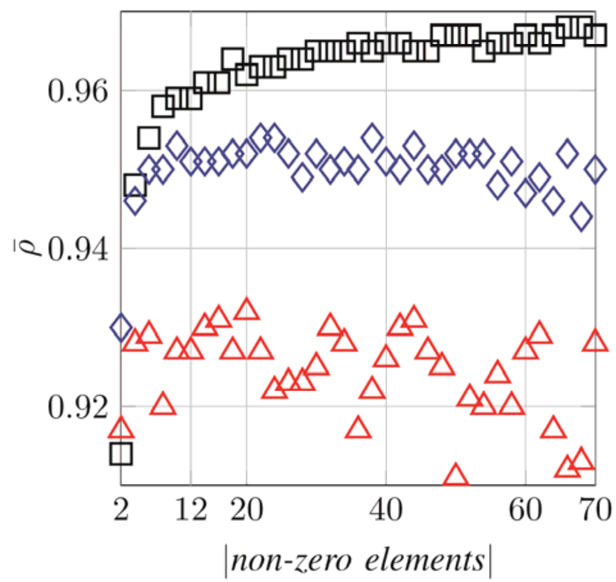
Experimental Results

- Construct the VSM using RMI method and repeat the sorting process.
  - Use different dimensionalities.
  - User different number of non-zero elements.
- Compare the sorted lists obtained from the VSM constructed at the original high dimension and the RMI-constructed VSM at reduced dimensionality.
  - Spearman's rank correlation (ρ) for comparison
- EXPECTATION: similar sorted lists from both VSM, i.e. $\rho = 1$.

Experimental Results

- Observed correlation:



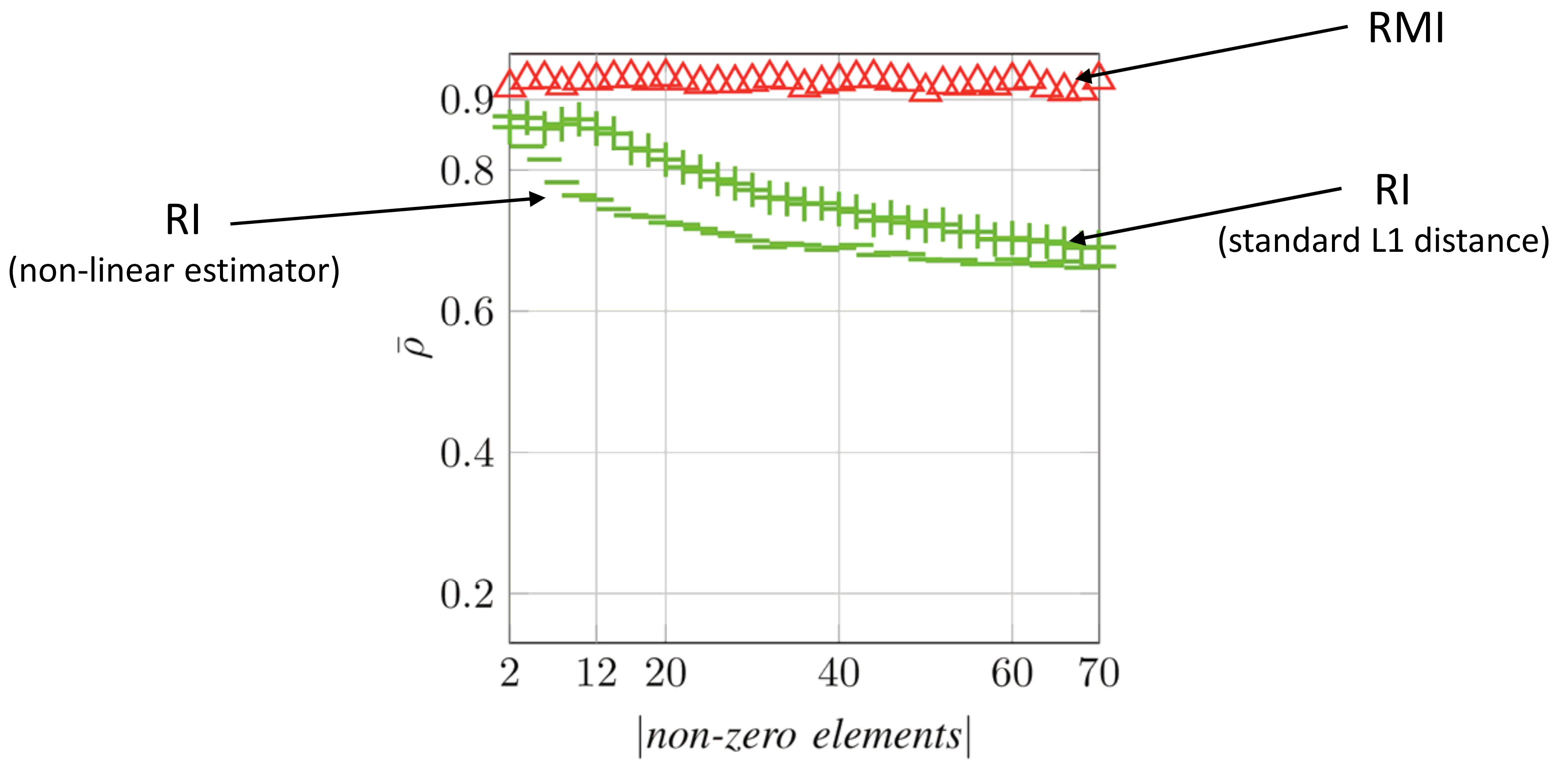


red triangles: dimension = 200
blue diamonds: dimension = 400
black squares: dimension = 800

For the baseline, we report ρ = 0.1375 for lists of documents sorted randomly.

Experimental Results

- Experiments support the earlier claim that RI do not preserve the L1 distance:



Discussion

- We proposed an incremental, scalable method for the construction of the L1 normed VSMs.
- We show the ability of the method in preserving the distances between documents.

- Is this possible to avoid floating-point calculation?
- What is the performance of RMI within the context of Information Retrieval benchmarks?

THANKS FOR YOUR ATTENTION!

Random Manhattan Indexing

Behrang Q. Zadeh and Siegfried Handschuh

National University of Ireland, Galway, Ireland
University of Passau, Lower Bavaria, Germany