ZigZag: Supporting Similarity Queries on Vector Space Models

WUHAN UNIVERSITY

University of California, Irvine

{ Wenhai Li and L.F. Deng and Y. Li }@whu.edu.cn Chen Li@uci.edu

Problem

The problem of similarity queries on vector space models is a general problem compared to commonly used functions such as Jaccard and cosine. It is a difficult problem on large-scale date sets due to three aspects.

1. Arbitrary token weights integrated in pair-wise similarity.
2. Variant degrees of each token in different records.
3. Given a threshold, exactly answer a query from millions of records.

Contributions

We formally define the problem of similarity queries on collections of records using a vector space model, and solve it efficiently based on a revised inverted index. Our main contributions are

1. Pruning methods with various bounds
2. Solid proofs of their correctness
3. An efficient elastic scheme to iteratively tighten the bounds
4. A family of algorithms based on an inverted index for large data sets
5. Evaluating intensive experiments on dozens million of records

Formal definition

Given a collection of records $\mathbb{S}$, a p-norm selection includes a record Q and a threshold τ. Suppose a token has a weight $w(t)$ and a degree $f(t,R)$ (in record R), it is to find all the records $R \in \mathbb{S}$ such that

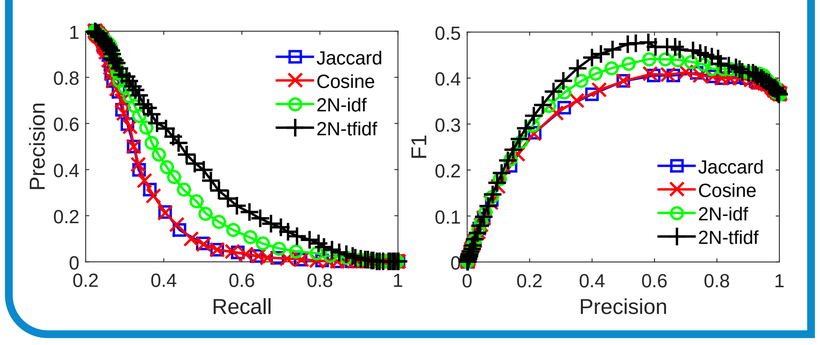
$$\frac{\left(\sum_{t\in R\cap Q}\left(f(t,R)w(t)\right)^{\frac{p}{2}}\left(f(t,Q)w(t)\right)^{\frac{p}{2}}\right)^{\frac{2}{p}}}{\left(\sum\limits_{t\in R}\left(f(t,R)w(t)\right)^{p}\sum\limits_{t\in Q}\left(f(t,Q)w(t)\right)^{p}\right)^{\frac{1}{p}}} \geq \tau$$

Usability

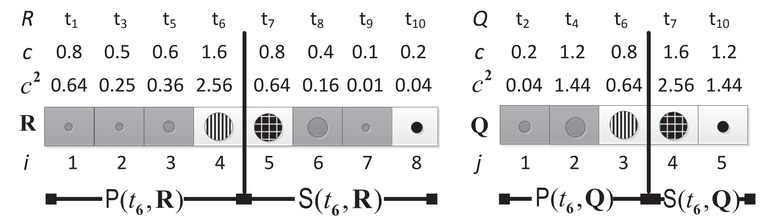
Based on the ground truth about similar pairs of records on "AskUbuntu", the two figures compare the precision and recall of this search using different functions. The results given above say that:

- The tf-idf 2-norm function gave the best accuracy results.
- The two 2-norm weighting functions were better than Jaccard and Cosine.

It was not to show that the p-norm function is the best for all applications. Instead, but a better similarity in certain applications.



Theoretic backgrounds



ℚ	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
$w(t)$	0.1	0.2	0.1	0.4	0.2	0.8	0.8	0.4	0.1	0.2
$f(t,R)$	8	0	5	0	3	**2**	**1**	1	1	**1**
$f(t,Q)$	0	1	0	3	0	**1**	**2**	0	0	**6**

Processed and remaining tokens of records.

We first define the processed and remaining tokens in a record when we consider all of its tokens based on a total order.

- The bound on remaining tokens is proposed to reduce the candidate size.
- The bounds on length and prefix can be used to save sequential IOs.
- Using the concepts in the figure we give solid proofs for all bounds.

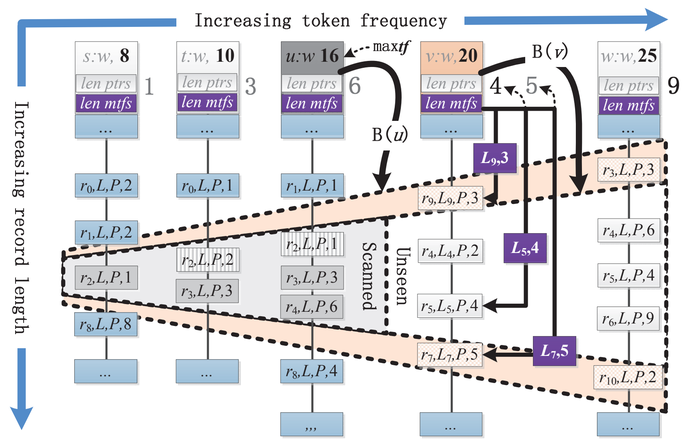
The inverted index is revised to support efficient pruning based on external storages.

- Associate token weights and degrees in the token and record entries.
- Support a skipping scheme and prefix in scanning the record lists of a token.

Elastic bounds

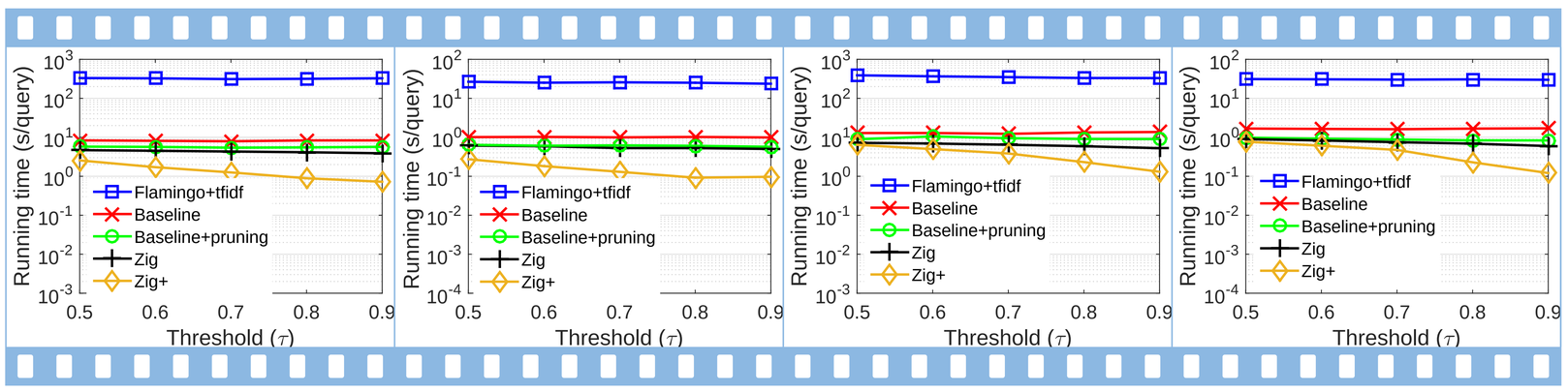
Significant improvement can be achieved as long as we iteratively tighten the bounds based on decreasing maximal degrees and reversely shrink the length bounds:

- After each step, the minimal and maximal degrees will be recomputed by the decreasing length bounded by the candidate set.
- The updated length bounds are used to shrink the range of the minimal and maximal degrees of unseen tokens.
- The maximal token degrees are associated in the index to iteratively tighten length bounds and maximal degrees until no more changes are observed.

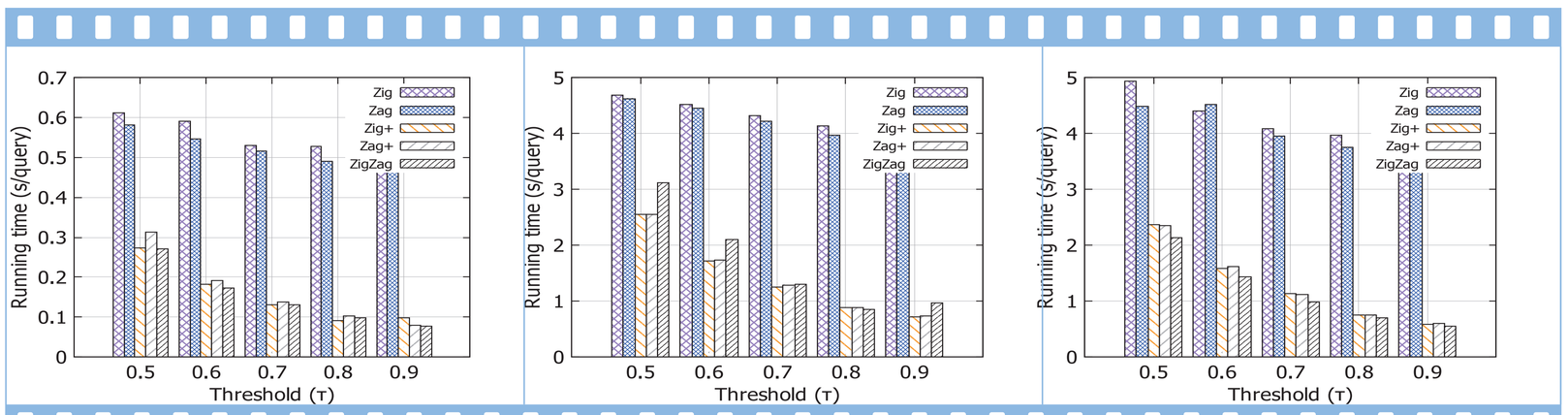


Scanning record entries using Elastic Bounds.

Results



MT on HDD | MT in Memory | Twitter on HDD | Twitter in Memory



MT in-memory | MT on HDD | MT on SSD

The comparisons with exiting methods demonstrated the superiority of the proposed schemes. The naïve Baseline is at least 10X faster than Flamingo, and Baseline+pruning is on average 2X better than Baseline. Zig+ is 1.2∼10X better than Zig.

ZigZag is the best one in the cases of using in-memory and SSD platforms, and falls short of the slow random IOs of HHD. Also, Zig+(Zag+) is always better than Zig(Zag).

A Future Direction

Can we associate part of the "remaining tokens" in the inverted index, allowing us to do "in-place" pruning. How to decide the associated length needs in-deep researches.

Source Code

Please contact by the following email or from Weechart by ⇒

Email: lwh@whu.edu.cn