Efficient Clustering from Distributions over Topics

Badenes-Olmedo, Carlos
Redondo Garcia, Jose Luís
Corcho, Oscar

Ontology Engineering Group (OEG)
Universidad Politécnica de Madrid (UPM)

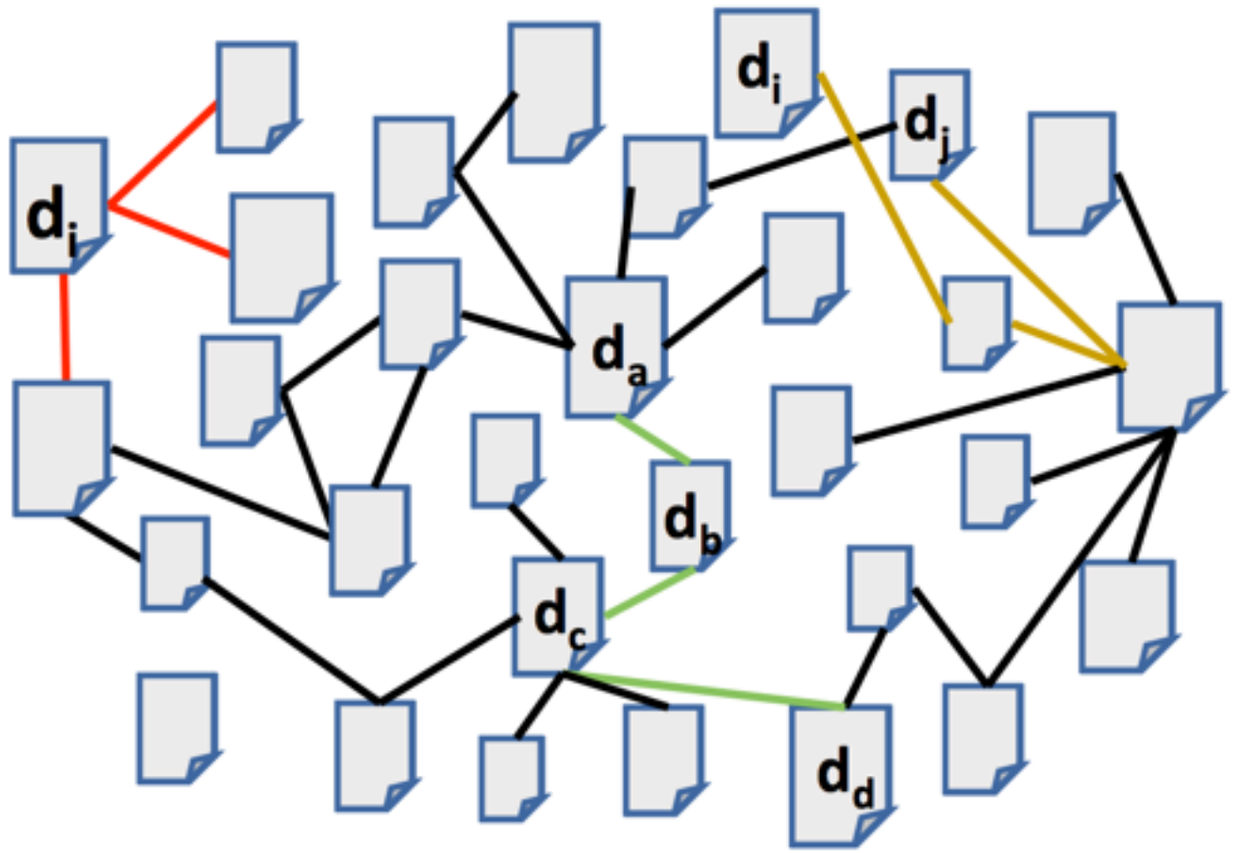
K-CAP 2017
Knowledge Capture
December 4th-6th, 2017
Austin, Texas, United States

cbadenes@fi.upm.es
@carbadol
oeg-upm.net
github.com/librairy

Textual documents everywhere

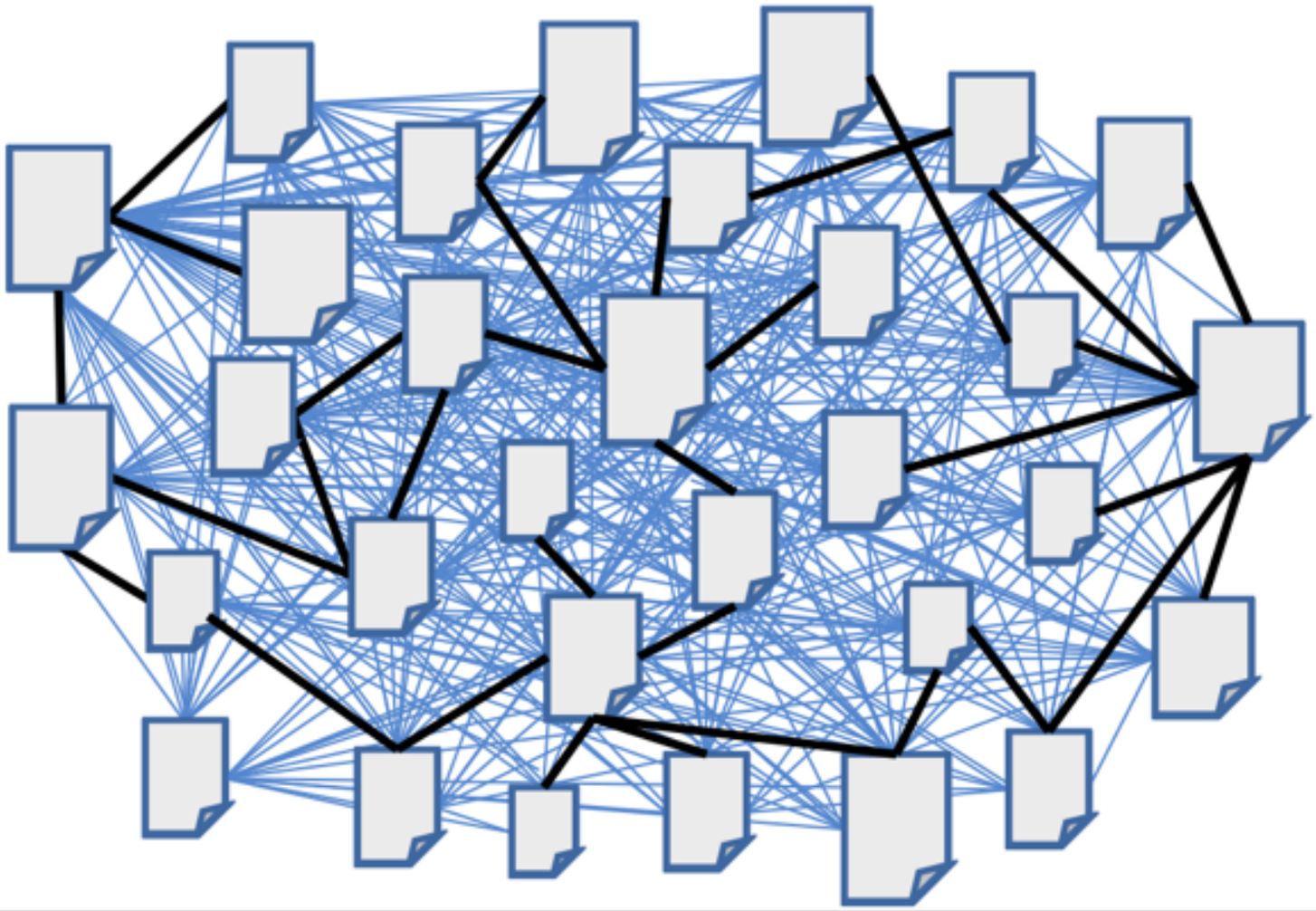
- Scientific papers
- Magazines
- Novels
- Social Media Posts
- Patents
- …

Connected Documents in a Collection



- Finding relations inside **big** collections of documents:
  - Identify related documents given a document d_i
  - Browse knowledge: from d_a to d_b to d_c …
  - Discover paths between documents: how to get from d_i to d_j

From Sets of Textual documents to Graphs



```
for d_i in Documents:
    for d_j in Documents:
        sim = Sim(d_i, d_j)
        if sim > threshold
            relatedDoc[d_i][d_j] = True
```

Similarity Matrix

$D = 31$

Big O = $O(N^2)$

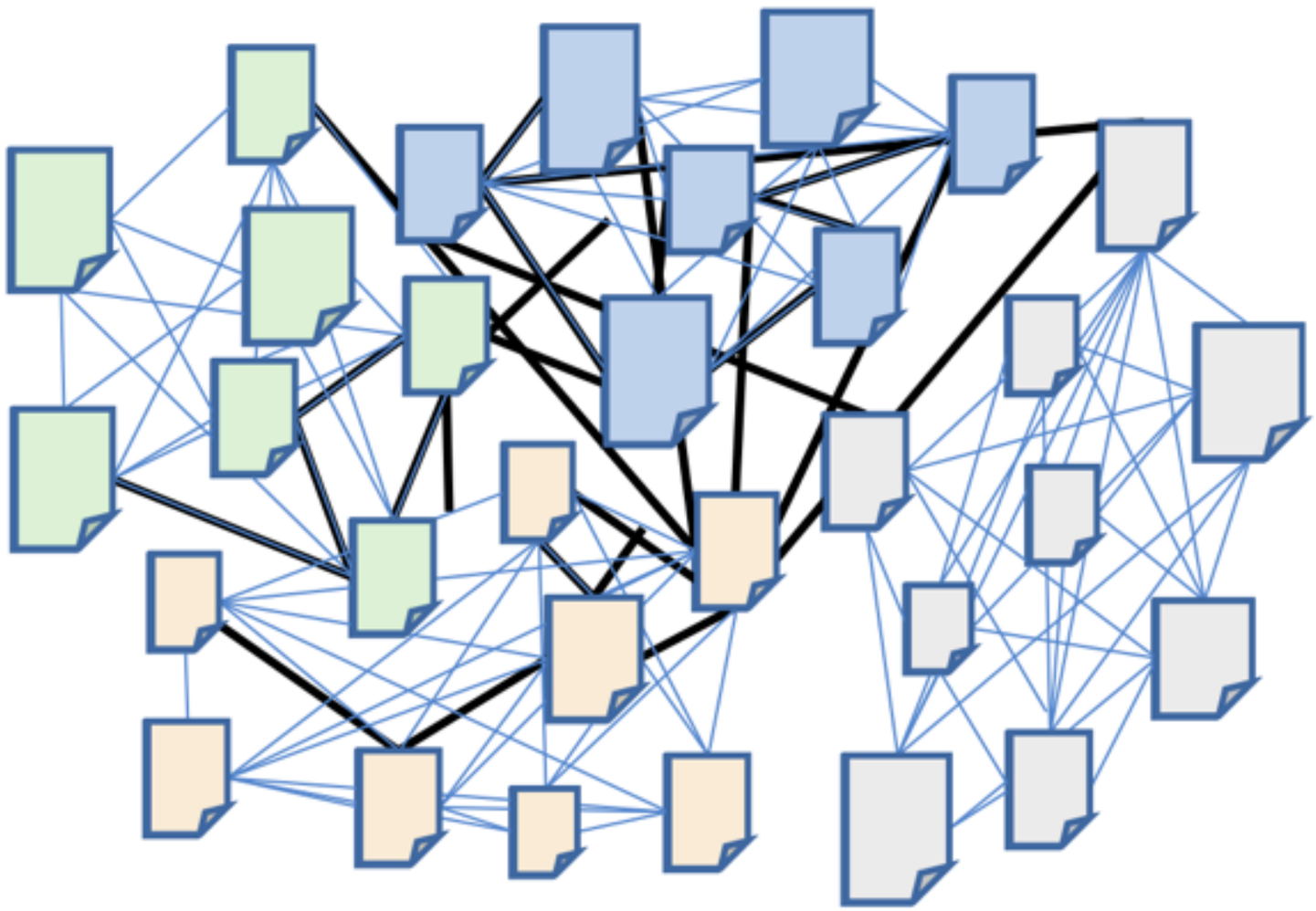
Non-Symmetric: 31*31-31=**930**

Symmetric: 930/2=**465**

Cost of $Sim(d_i, d_j)$??

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	1	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	**1**	0	1	0	0	0	0	**1**	0	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	**1**	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0	0	0	0	**1**	0	0	0
6	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	1	0	0	**1**	0	0	0	0	**1**	0	0	0	0	0	0	0	0	**1**	0	0	0	0	0	0	0
8	0	0	**1**	0	0	0	0	1	0	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	**1**	0	0	0	0	0
9	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	**1**	0	0	0	0	**1**	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0	0	**1**	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	1	0	**1**	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0
15	0	0	**1**	**1**	0	0	**1**	**1**	0	0	0	0	**1**	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	**1**	0	0	**1**	0	0	**1**	**1**	**1**
19	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	**1**	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
23	0	0	0	0	**1**	0	0	0	0	**1**	**1**	**1**	0	**1**	0	0	0	**1**	0	0	0	0	1	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
26	0	0	**1**	0	0	0	0	**1**	0	**1**	0	0	**1**	0	0	0	0	**1**	0	**1**	0	0	0	0	0	1	0	0	0	**1**	0
27	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
28	0	0	0	0	**1**	0	0	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	1	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0	0	0	0	0	0	0	**1**	0	0	0	1	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	1

Solution: Partition of the Search Space



- ↓Temporal Cost
- ↑ Precision
- ↓ Recall.

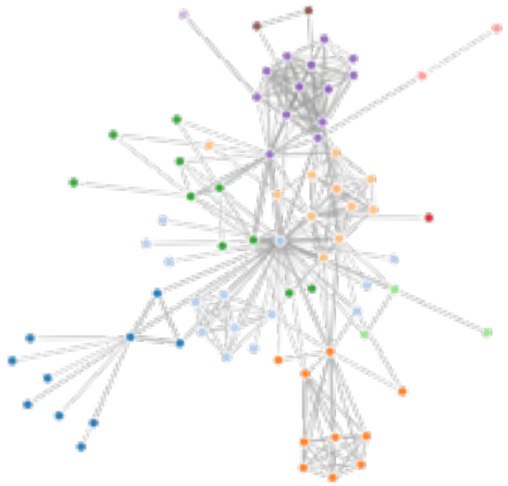
1. Partitioning the collection into different groups **C**:
2. Calculating similarity **only** over the documents $\mathbf{d_c}$ inside **C**

Use-Case: Digital Publisher

All pairwise similarities

4x1.6 Ghz
16GB RAM

4x1.6 Ghz
16GB RAM



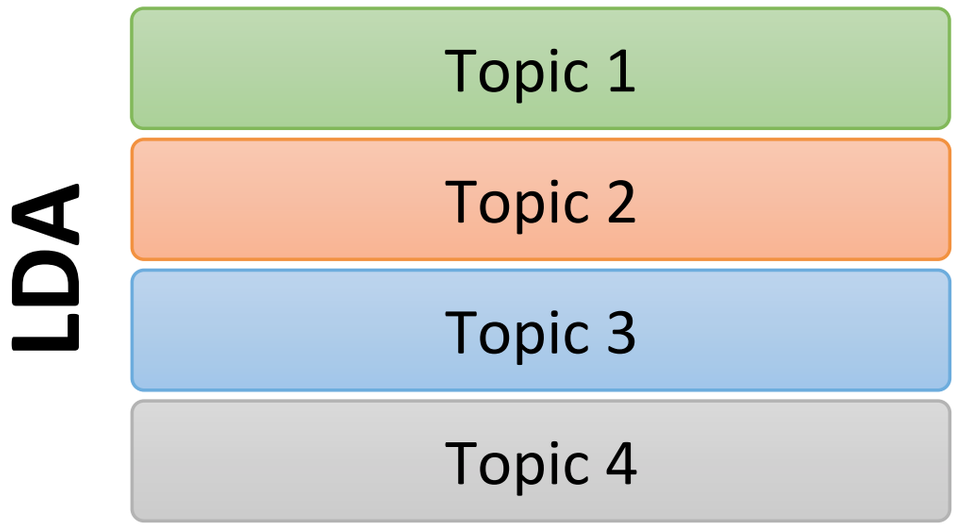
7,648	Books
68,653	Chapters
76,301	Documents

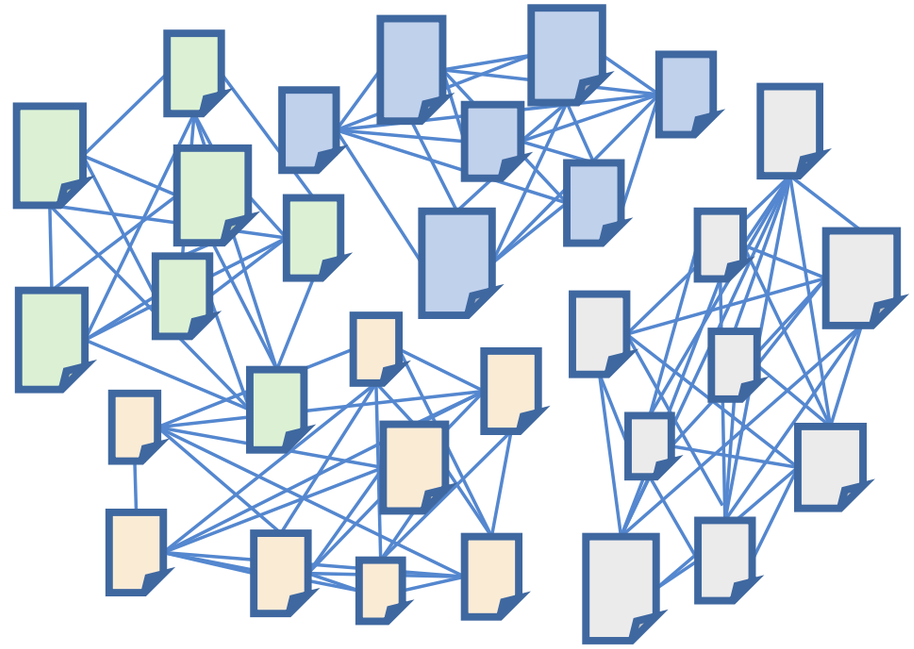
2,910,883,150
similarities

aprox 8 hours

LDA for efficient space partition

Probabilistic Topic Models (PTM) and in particular, on Latent **Dirichlet Allocation (LDA)** can **efficiently** divide the search space and speed up the process of finding relations among documents inside big collections.





Probabilistic Topic Models

Topics

gene	0.04
dna	0.02
genetic	0.01
...	

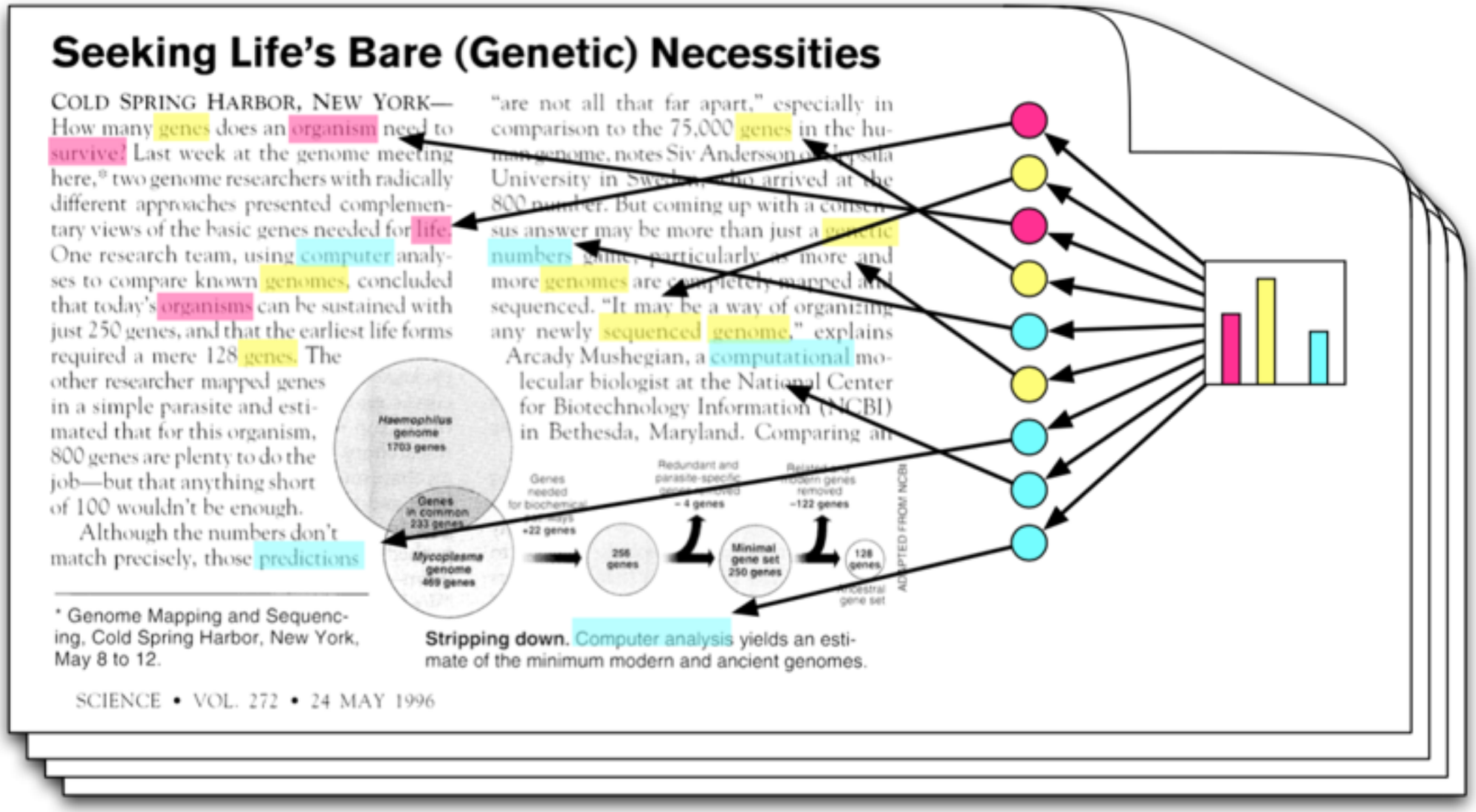
life	0.02
evolve	0.01
organism	0.01
...	

brain	0.04
neuron	0.02
nerve	0.01
...	

data	0.02
number	0.02
computer	0.01
...	

Documents

Topic proportions and assignments



Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predictions "are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an

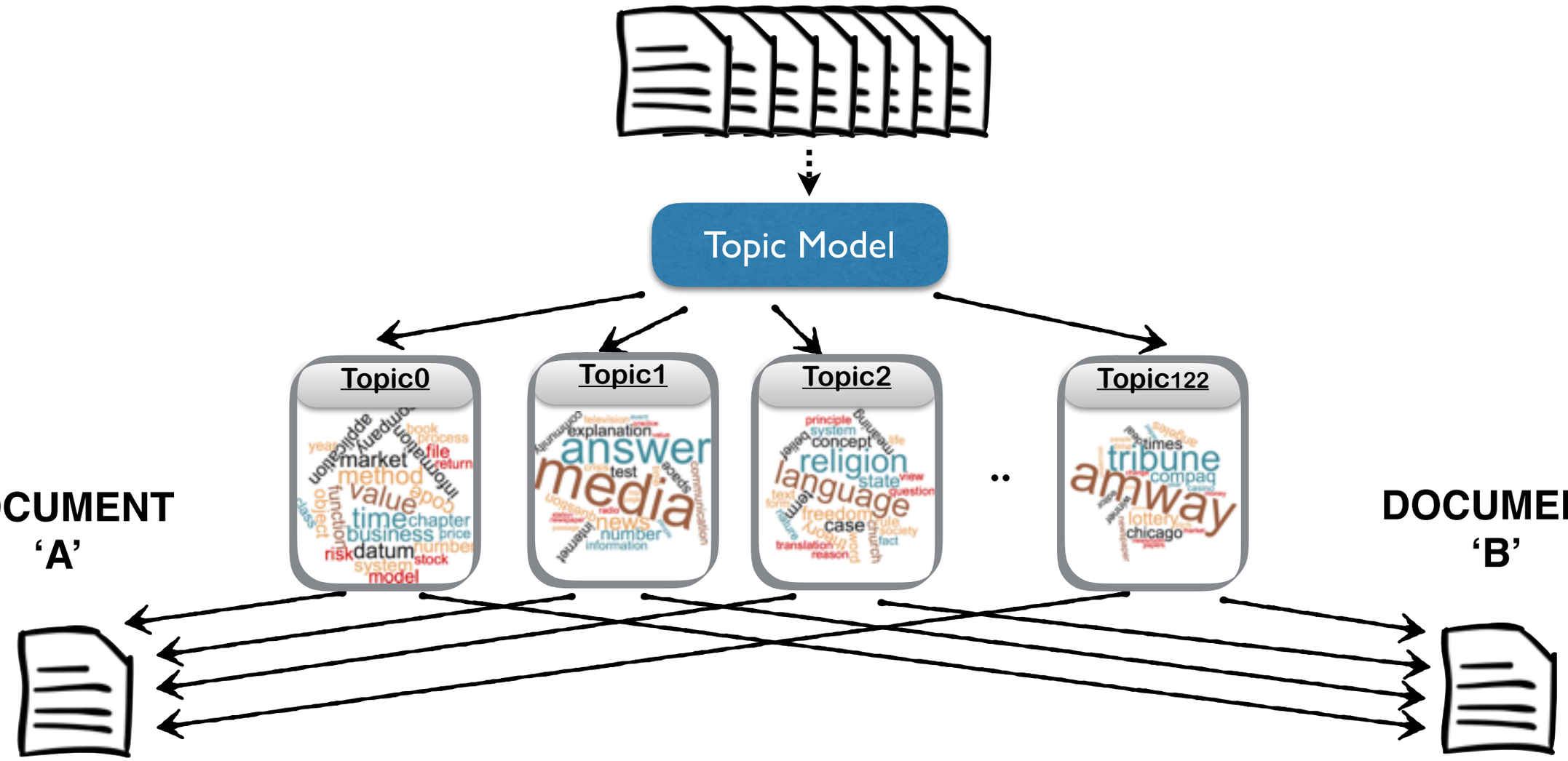
* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

SCIENCE • VOL. 272 • 24 MAY 1996

- Each **topic** is a distribution over words
- Each **document** is a mixture of corpus-wide topics
- Each **word** is drawn from one of those topics

Topic-based Similarity



topic0: 0.520,
topic1: 0.327
topic2: 0.081
…
topic122: 0.182

Similarity Value

0.595

topic1: 0.573,
topic2: 0.172
topic3: 0.136
…
topic122: 0.099

Jensen-Shannon Divergence

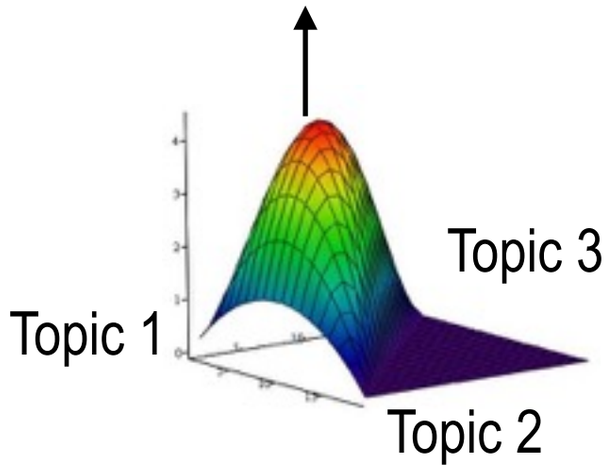
$$JSD(p,q) = \sum_{i=1}^{T} p_i * \log \frac{2 * p_i}{p_i + q_i} + \sum_{i=1}^{T} q_i * \log \frac{2 * q_i}{q_i + p_i}$$

Scenario

corpus

Prob. Topic Model

[0.243, 0.145, 0.600, 0.022]



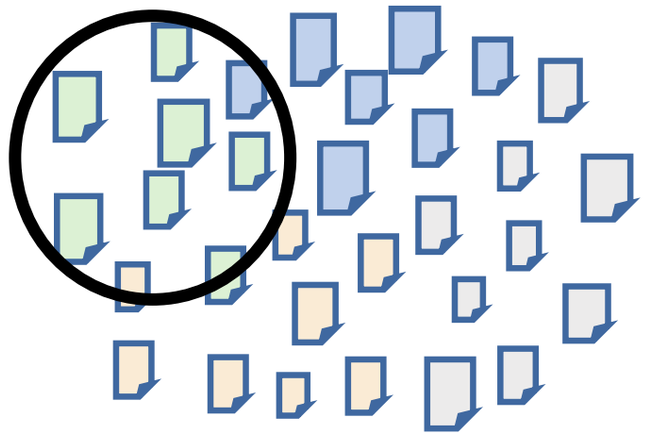
Dirichlet Distribution

- Exponential family distribution over the simplex, i.e. ***positive vectors that sum to one***

Trends on Dirichlet-based Clustering (TDC)

- Instead of directly relying on the topic distribution's scores, it considers their variations across consecutive topics inside a document's topic distribution

$$\boldsymbol{TDC(P) = T} \qquad \textit{Where:}$$
$$\boldsymbol{T_i = 1}, \textit{when } \boldsymbol{P_i < P_{i+1}} \textbf{ (upward)}$$
$$\boldsymbol{T_i = 2}, \textit{when } \boldsymbol{P_i > P_{i+1}} \textbf{ (downward)}$$
$$\boldsymbol{T_i = 0}, \textit{when } \boldsymbol{P_i = P_{i+1}} \textbf{ (neutral)}$$



T =

2	1	2	0
>	<	>	=

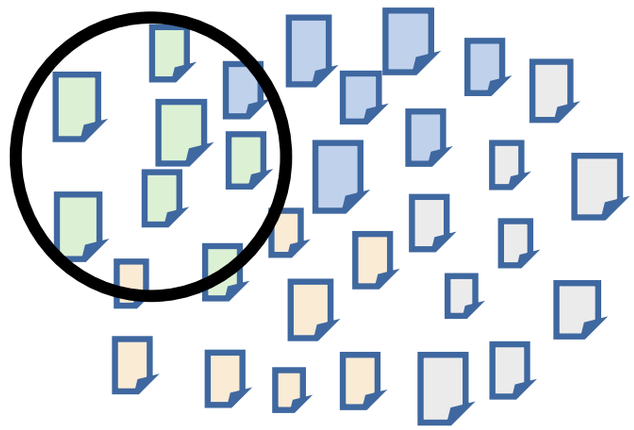
P =

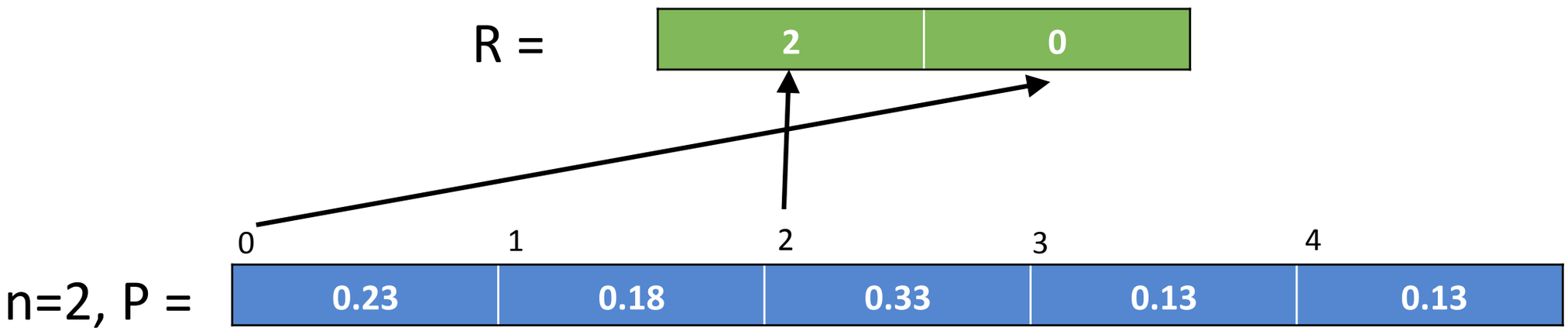
0.23	0.18	0.33	0.13	0.13

Ranking on Dirichlet-based Clustering (RDC)

- Only considering the **top n** topics from the ranked list of probability distributions [29]
- Based on the assumption that the highest weighted topics have a high influence in the rest of topics when calculating distances

$$\mathbf{RDC}(\boldsymbol{P},\boldsymbol{n}) = \boldsymbol{R} \qquad Where:$$
$$\forall j \in r, i \in, P \;\; P_{R_j} \geq P_{R_{[j-1]}}, and\ P_{R_j} \geq P_k, \forall k \notin R$$



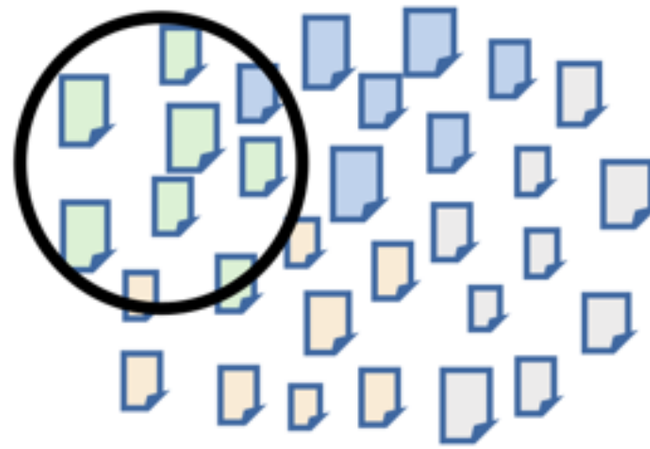


Cumulative Ranking on Dirichlet-based Clustering (CRDC)

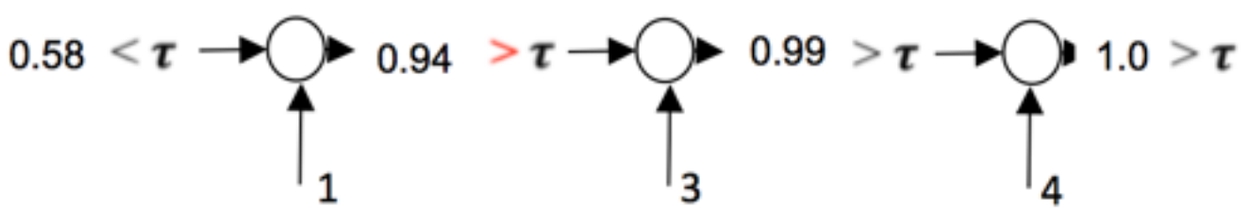
- Only considering the **top n** topics until the sum of the weights of the highest topics exceeded a given threshold

$$\mathbf{CRCD}(P, \tau) = C \qquad Where:$$

$$\forall j = c,\ P_{C_j} \geq P_{C_{[j-1]}}, and \sum_{l=0}^{c} P_{C_l} > \tau$$



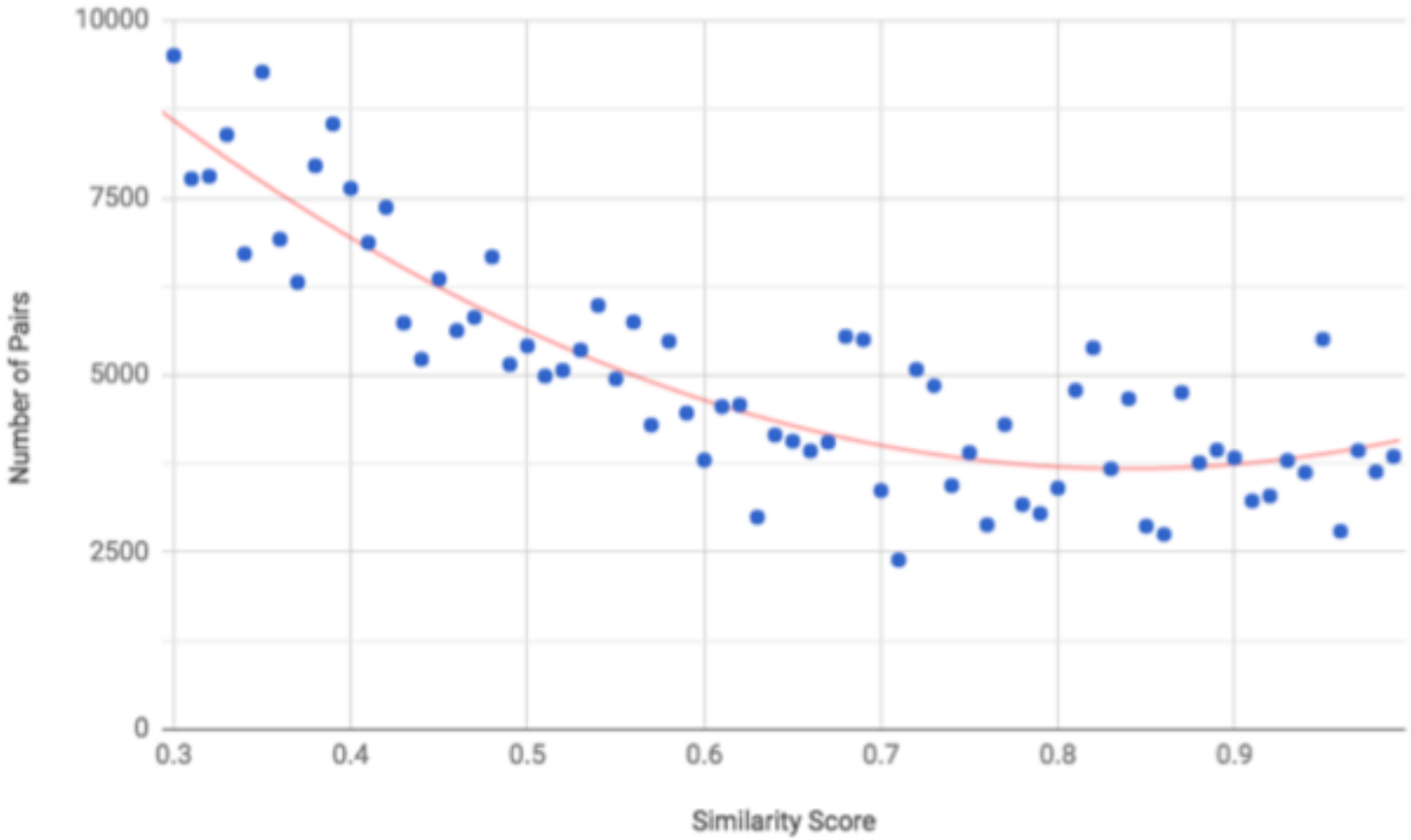
R = [2 | 1 | 3 | 4]



τ =0.9, P= [1: 0.36 | 2: 0.58 | 3: 0.05 | 4: 0.01]

Experiments: Dataset

- **Sources**
  - DIRICHLET-RANDOM-MIXTURE (DRM)
  - Papers published in the Advances in Engineering Software (AIES)



- **Num of documents**: 1000
- **LDA Configuration [28]**

$$\alpha = \frac{50}{k},\ \beta = 0.01,\ k = 2 * \sqrt{\frac{n}{2}}$$

- **Threshold**: 0.83

Clustering Algorithms:

- **K-Means** as a centroid-based clustering approach
  - implemented at the Apache Commons Math[1]
  - k=44, maxIterations=50
- **DBSCAN** as a density-based clustering approach
  - eps = 0.1 and minPts = 50
- **Random**, which randomly selects R from the dataset
  - randomly divides the dataset into m equal-sized groups
- RDC, n = 1
- CRDC, τ is set to 0.9.

Similarity Metrics:

- Jensen-Shannon Divergence
- Hellinger Distance

1: http://commons.apache.org/proper/commons-math/

Experiments: Measures

- **cost**: number of similarity calculations required by the algorithm:

$$Cost = \frac{(reqSim - minSim)}{(totalSim - minSim)}$$

minSim: number of similar documents in Gold Standard

totalSim: Cartesian product of existing documents: (1000^2 = 1m)

- **Effectiveness**: based on precision and recall. It expresses the quality of the algorithm:

$$\text{effectiveness} = \frac{(precision^2 * recall^2)}{2}$$

- **Efficiency:** compromise between quality and performance:

$$\text{efficiency} = \text{effectiveness} - \text{cost}$$

Precision

Size	CRDC	DBSCAN	K-Means	RDC	TDC	Random
200	0.94	0.10	**0.96**	0.31	0.42	0.12
300	0.93	0.15	**0.94**	0.30	0.39	0.08
400	**0.93**	0.15	0.89	0.29	0.39	0.09
500	**0.92**	0.30	0.90	0.28	0.38	0.09
600	**0.92**	0.19	0.88	0.28	0.38	0.08
700	**0.92**	0.20	0.91	0.28	0.38	0.09
800	**0.92**	0.12	0.89	0.30	0.39	0.10
900	**0.92**	0.13	0.87	0.30	0.40	0.10
1000	**0.93**	0.13	0.90	0.30	0.40	0.10

Table 1: Precision (JS-based) in AIES

Size	CRDC	DBSCAN	K-Means	RDC	TDC	Random
200	0.75	0.07	**0.84**	0.23	0.08	0.33
300	0.74	0.08	**0.83**	0.23	0.06	0.32
400	**0.76**	**0.09**	0.76	0.22	0.06	0.32
500	0.73	0.08	**0.74**	0.21	0.08	0.31
600	0.72	0.08	**0.73**	0.21	0.06	0.30
700	0.71	0.10	**0.76**	0.21	0.06	0.30
800	0.73	0.11	**0.78**	0.22	0.07	0.31
900	0.73	0.12	**0.80**	0.22	0.08	0.32
1000	0.74	0.15	**0.77**	0.23	0.08	0.32

Table 2: Precision (He-based) in AIES

Recall

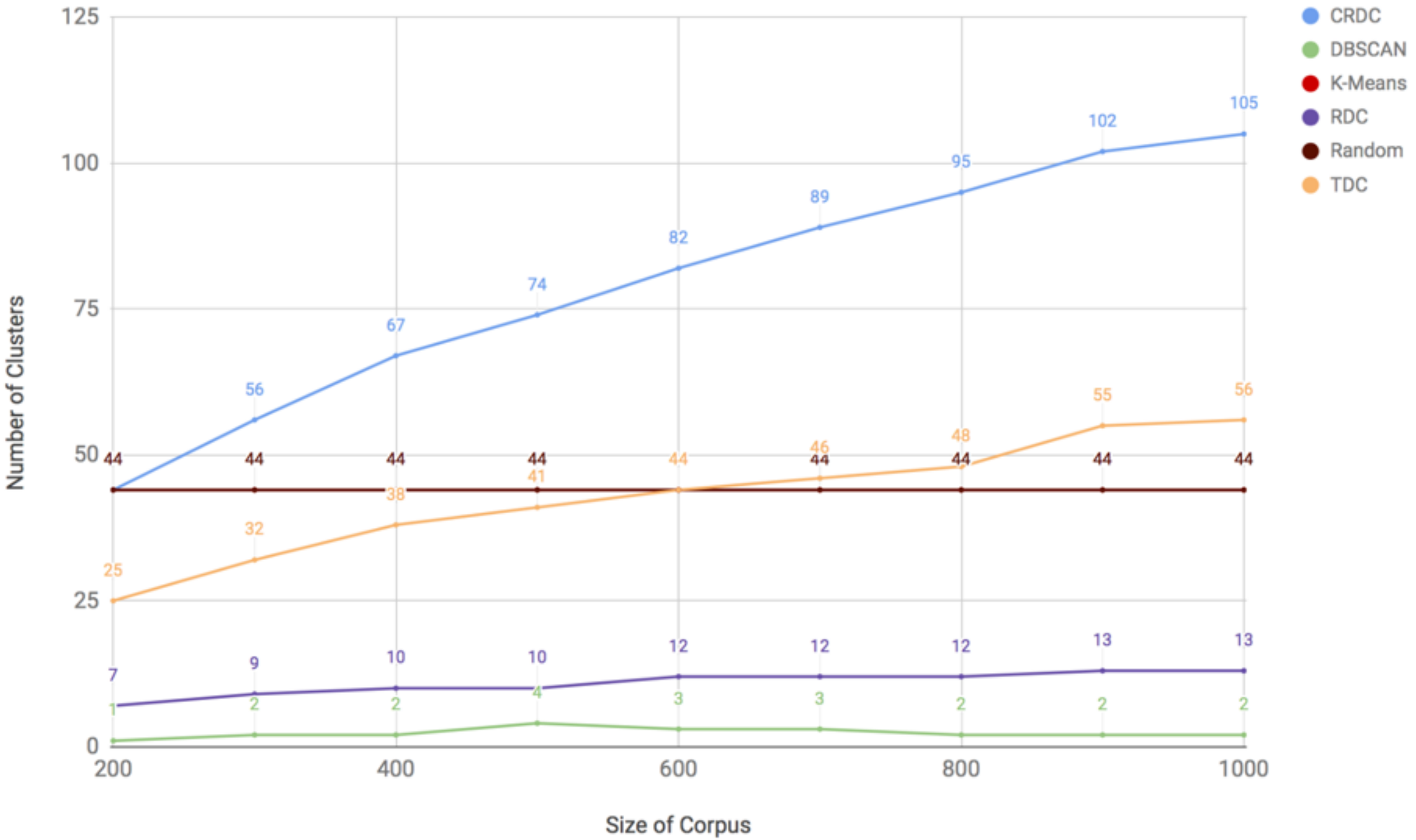
Size	CRDC	DBSCAN	K-Means	RDC	TDC	Random
200	0.92	**1.00**	0.79	0.96	0.02	0.87
300	0.91	0.89	0.84	**0.96**	0.02	0.84
400	0.92	0.92	0.90	**0.96**	0.02	0.86
500	0.91	0.94	0.88	**0.96**	0.03	0.85
600	0.91	0.94	0.87	**0.96**	0.02	0.83
700	0.91	0.92	0.90	**0.96**	0.02	0.83
800	0.92	0.92	0.88	**0.96**	0.02	0.83
900	0.92	0.95	0.86	**0.96**	0.02	0.83
1000	0.92	0.93	0.89	**0.97**	0.02	0.84

Table 3: Recall (JS-based) in AIES

Size‘	CRDC	DBSCAN	K-Means	RDC	TDC	Random
200	0.84	**1.00**	0.65	0.96	0.02	0.82
300	0.84	**0.98**	0.76	0.95	0.02	0.78
400	0.84	**0.98**	0.79	0.94	0.02	0.79
500	0.85	0.94	0.87	**0.95**	0.02	0.78
600	0.86	**0.96**	0.80	0.95	0.02	0.76
700	0.85	**0.98**	0.80	0.95	0.02	0.76
800	0.85	**0.99**	0.81	0.95	0.02	0.76
900	0.85	**0.99**	0.75	0.95	0.02	0.77
1000	0.86	**1.00**	0.74	0.96	0.02	0.78

Table 4: Recall (He-based) in AIES

Number of Clusters

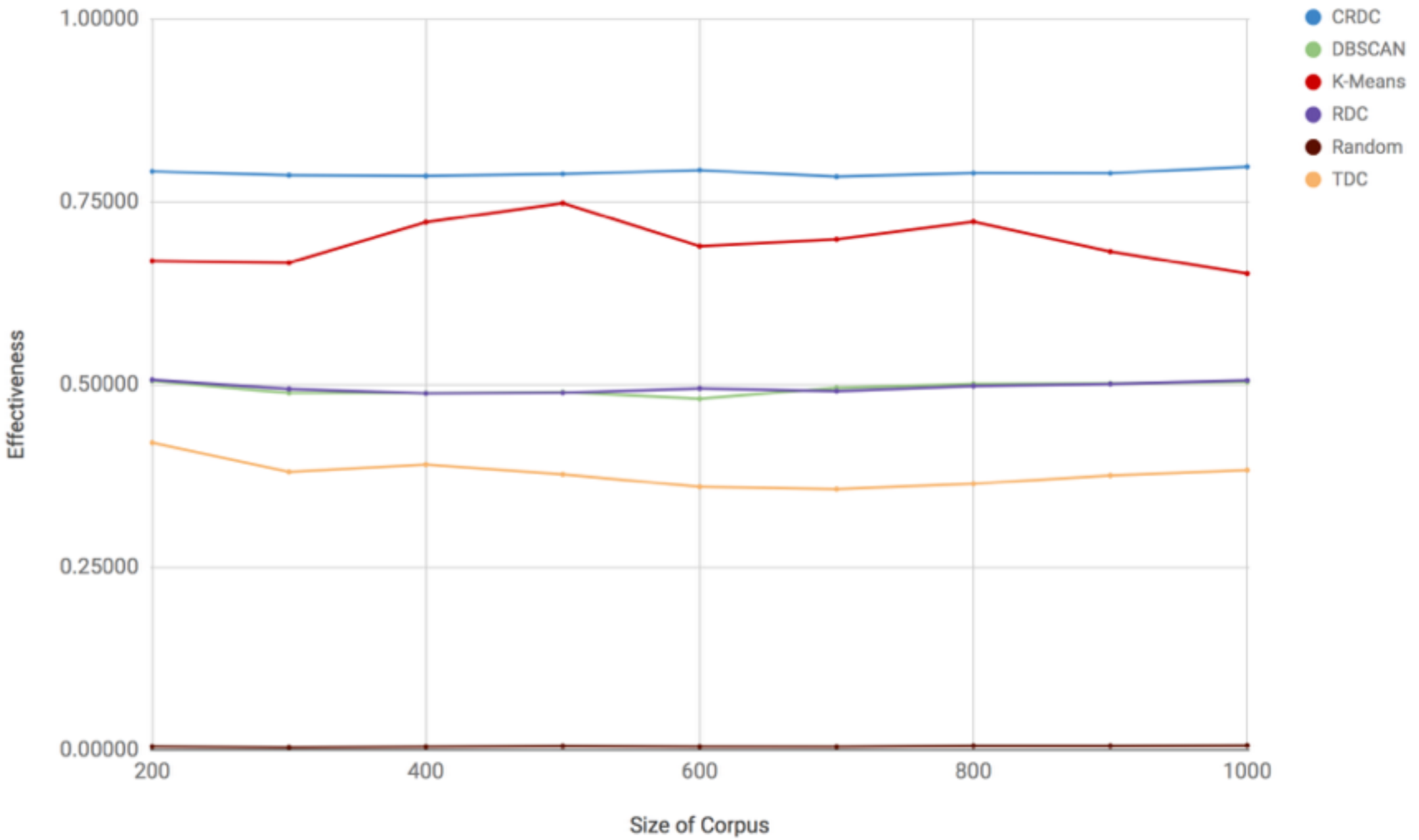


Experiments: Results

Effectiveness

$$\text{effectiveness} = \frac{(precision^2 * recall^2)}{2}$$

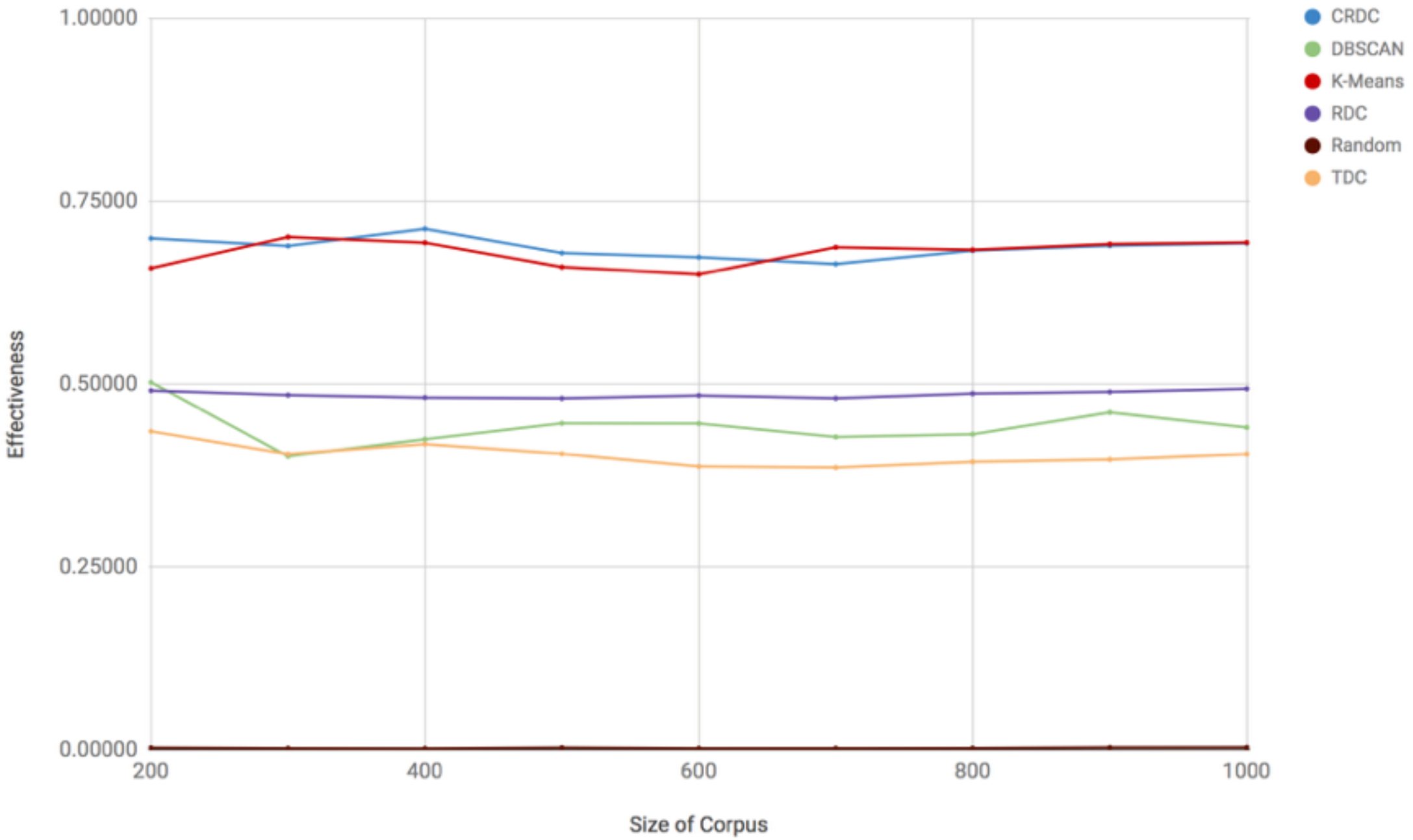
JSD



Effectiveness

$$\text{effectiveness} = \frac{(precision^2 * recall^2)}{2}$$

Hellinger

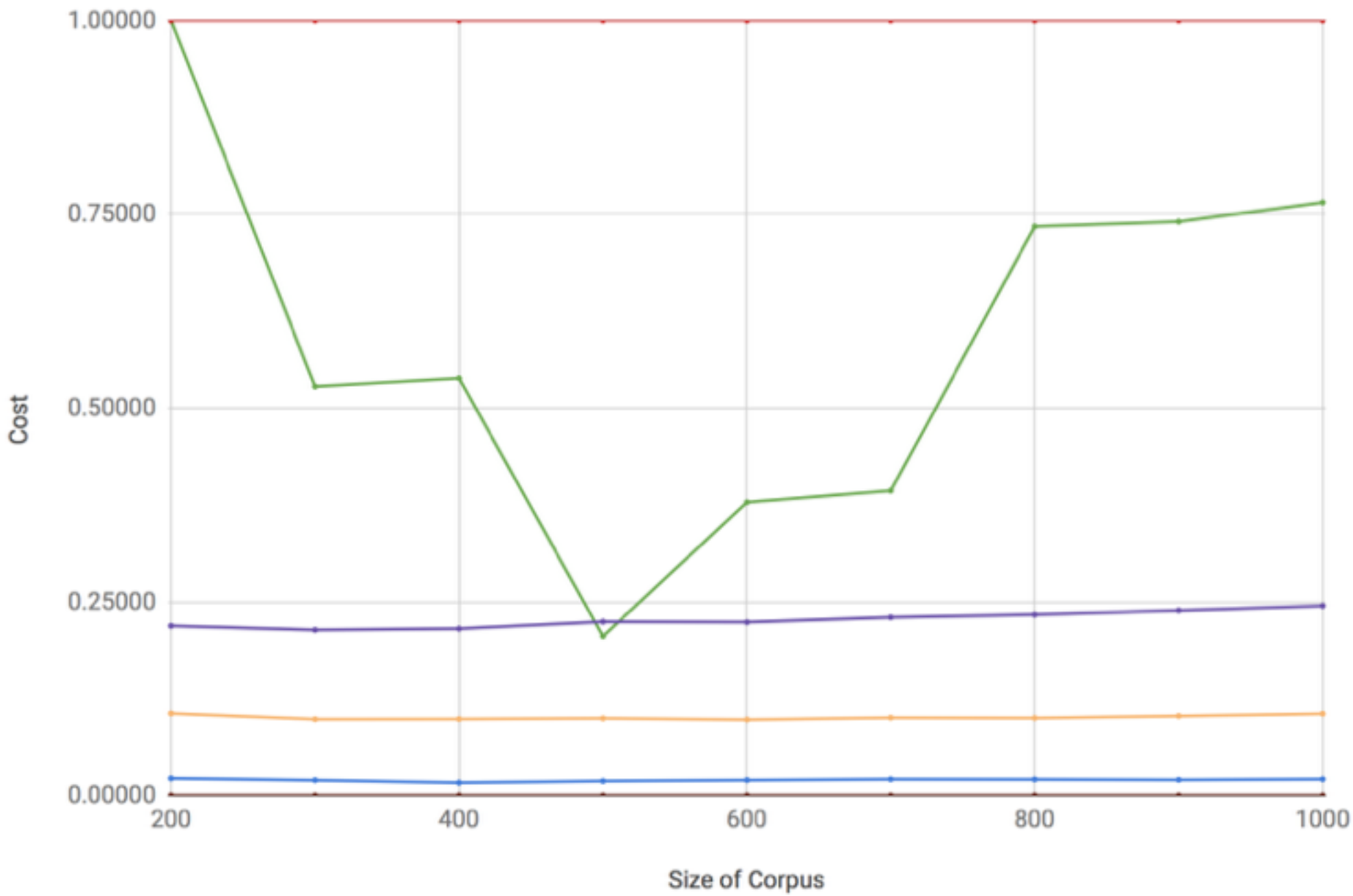


Experiments: Results

Cost

$$Cost = \frac{(reqSim - minSim)}{(totalSim - minSim)}$$

JSD

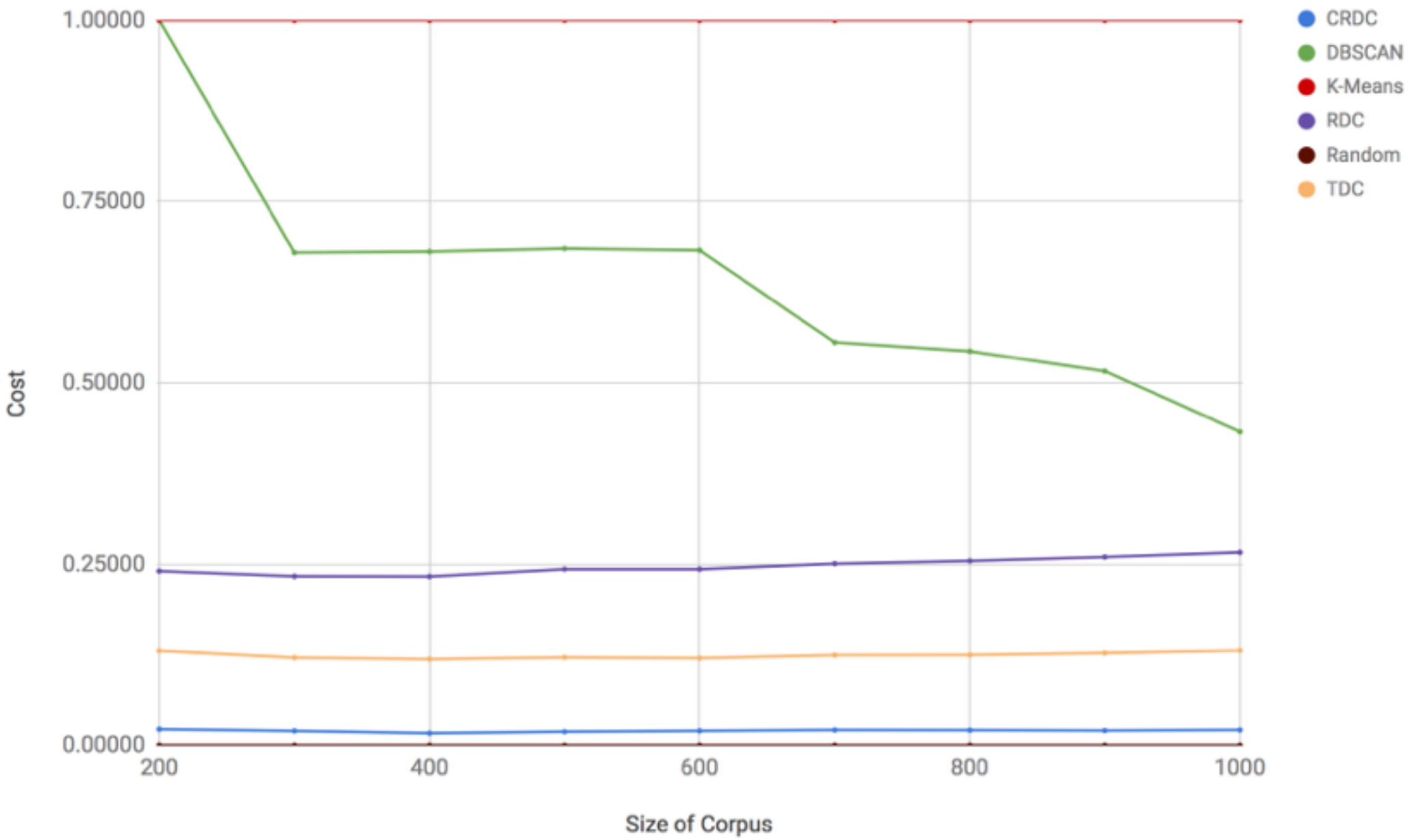


Experiments: Results

Cost

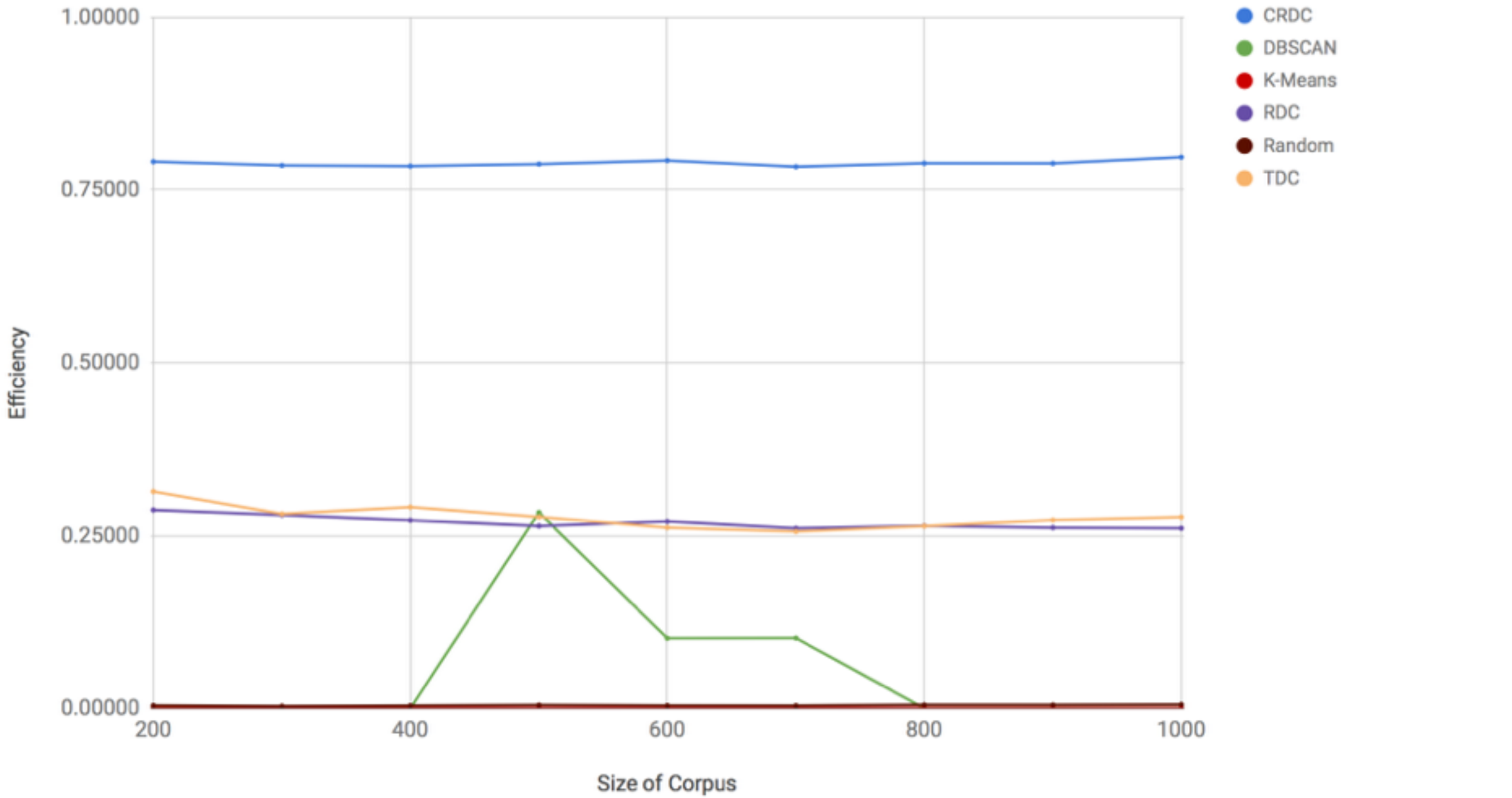
$$Cost = \frac{(reqSim - minSim)}{(totalSim - minSim)}$$

Hellinger



Efficiency

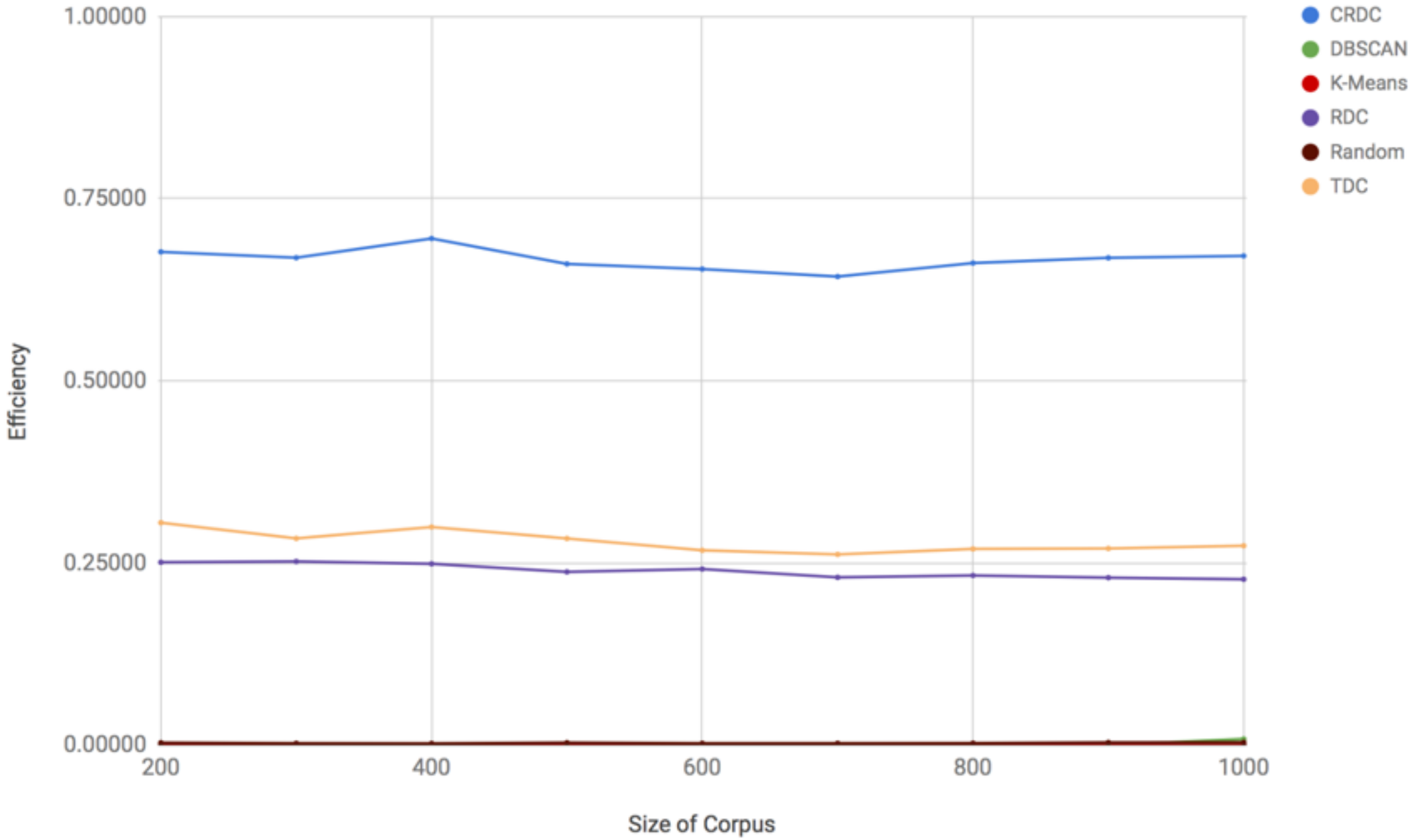
$$\text{efficiency} = \text{effectiveness} - \text{cost}$$



Efficiency

$$\text{efficiency} = \text{effectiveness} - \text{cost}$$

Hellinger



Conclusions

- Unsupervised clustering algorithms, TDC, RDC and CRDC.
- CRDC is a promising approach, which improves the efficiency obtained by other centroid-based and density-based approaches such as K-Means
- Hierarchical approach for RDC algorithm was also considered but it did not produce good results

Future Work

- Hybrid methods combining some of these novel approaches with existing techniques will be performed in future work on the same line
- Nearest neighbors as baseline

Use-Case: Digital Publisher

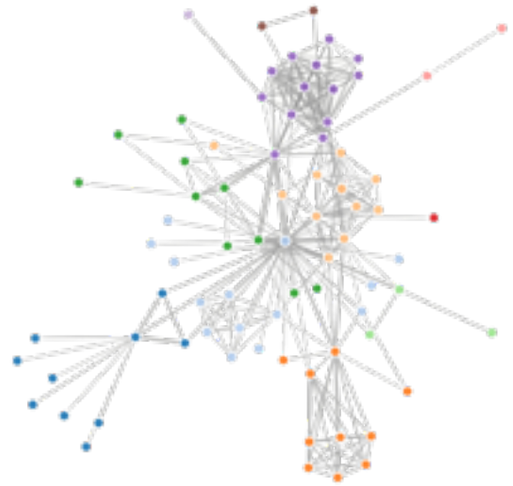
by using CRDC

7,648 Books
68,653 Chapters

76,301 Documents

4x1.6 Ghz
16GB RAM

4x1.6 Ghz
16GB RAM



~~2,910,883,150~~
similarities

~~*aprox* 8 hours~~

aprox 2 hours

64,635,080
similarities

threshold = 0.9
accuracy 0.81

Efficient Clustering from Distributions over Topics

Badenes-Olmedo, Carlos
Redondo Garcia, Jose Luís
Corcho, Oscar

Ontology Engineering Group (OEG)
Universidad Politécnica de Madrid (UPM)

cbadenes@fi.upm.es
@carbadol
oeg-upm.net
github.com/librairy

K-CAP 2017
Knowledge Capture
December 4th-6th, 2017
Austin, Texas, United States