Advanced tf weighting schemes, ranking and Graph of Words

Michalis Vazirgiannis
LIX @ Ecole Polytechnique

January 2017

Outline

- **Advanced tf weighting schemes**
- Graph-of-words

Term Frequency (1)

- **term frequency** $tf(t,d)$,
  - simplest choice: use the *raw frequency* of a term in a document, i.e. $tf(t,d) = f(t,d)$: the number of times that term t occurs in document d:
  - Other possibilities:
- boolean "frequencies": $tf(t,d) = 1$ if t occurs in d and 0 otherwise;
- logarithmically scaled frequency: $tf(t,d) = 1 + \log f(t,d)$ (and 0 when $f(t,d) = 0$)
- normalized frequency, $$tf(t,d) = \frac{f(t,d)}{\max\{f(w,d) : w in d\}}$$
  - raw frequency divided by the maximum raw frequency of any term in the document.
  - prevent a bias towards longer documents,.

Term frequency: Log frequency weighting

- The log frequency weight of term t in d is defined as follows

$$w_{t,d} = \begin{cases} 1 + \log_{10} \text{tf}_{t,d} & \text{if } \text{tf}_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}$$

- $tf_{t,d} \rightarrow w_{t,d}$: 0 → 0, 1 → 1, 2 → 1.3, 10 → 2, 1000 → 4, ...
- Score for a query document (q,d) pair:

$$tf(q, d) = \sum_{t \in q \cap d} (1 + \log tf_{t,d})$$

- The score is 0 if none of the query terms is present in the document.

Document frequency

- high weights for rare terms like Hypermnesia
- low (positive) weights for frequent words like GOOD, INCREASE and LINE.
- Factor document frequency into the matching score.
- The document frequency df_t is # of documents in the collection that the term occurs in.
- df_t is an inverse measure of the informativeness of term t.
- We define the idf weight of term t as: $\mathrm{idf}_t = \log_{10} \frac{N}{\mathrm{df}_t}$

  (*N*: # documents in the collection.)
- idf_t is a measure of the informativeness of the term.
- $[\log N/df_t]$ instead of $[N/df_t]$ to "dampen" the effect of *idf*

tf-idf weighting

- The tf-idf weight of a term is the product of its tf weight and its idf weight.

$$w_{t,d} = (1 + log tf_{t,d}) * log \frac{N}{df_t}$$

- increases with the number of occurrences within a document. (tf)
- increases with the rarity of the term in the collection. (idf)
- Best known weighting scheme in information retrieval

Term weighting in VSM

- Term weighting with tf-idf (and variations)

$$w_{t,d} = tf_{t,d} \times idf_t$$

- tf models the importance of a term in a document

$$tf_{t,d} = f_{t,d} \qquad\qquad tf_{t,d} = \frac{f_{t,d}}{\max\left(f_{s,d}\right)}$$

  – $f_{t,d}$ is the frequency of term *t* in document *d*
- idf models the importance of a term in the document collection
  – Logarithm base not important
  – Information content of event "term *t* occurs in document *d*"

$$idf_t = -\log P\left(t \text{ occurs in } d\right) = -\log\frac{n_t}{N} = \log\frac{N}{n_t} \qquad\qquad idf_t = \log\frac{N - n_t + 0.5}{n_t + 0.5}$$

  – *N* is the total number of documents, n_t is the document frequency of term *t*

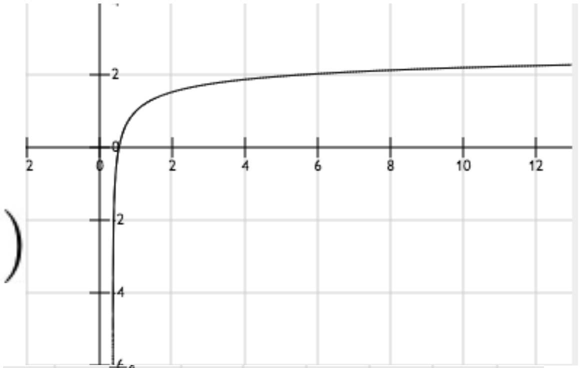
TF normalizations

- Fang *et al.* proposed a set of **heuristic retrieval constraints** for scoring functions to regularize the term frequency (tf) and the document frequency (df) in ad hoc IR.
- Research community developed a set of corresponding **normalization functions** to apply on tf (concave functions, document length normalization...)
- Intuitively, **the more times a document contains a query term** in ad hoc IR, **the more relevant this document is for the query**.
- Mathematically, this corresponds to a retrieval model that is an **increasing function of the term frequency**.
- However, use of raw term frequency proved to be non-optimal in ad hoc IR,
- Research community started normalizing the tf considering multiple heuristic retrieval constraints translated into mathematical functions to apply on top of the term frequency.

Tf concave function

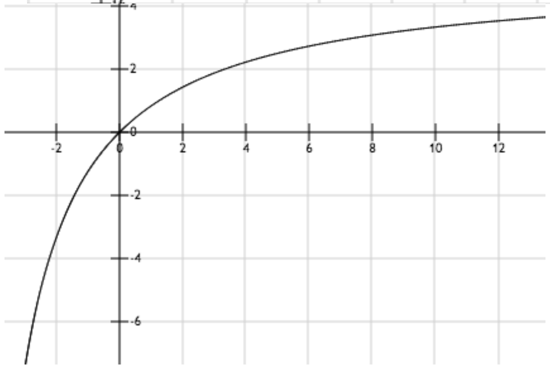
- The **marginal gain in relevance** of seeing an additional occurrence of a term inside a document is **decreasing**.
- Mathematically, this corresponds to applying a concave function:
  - Log –concavity

$$TF_l(t,d) = 1 + ln(1 + ln(tf(t,d)))$$



  - K-concavity

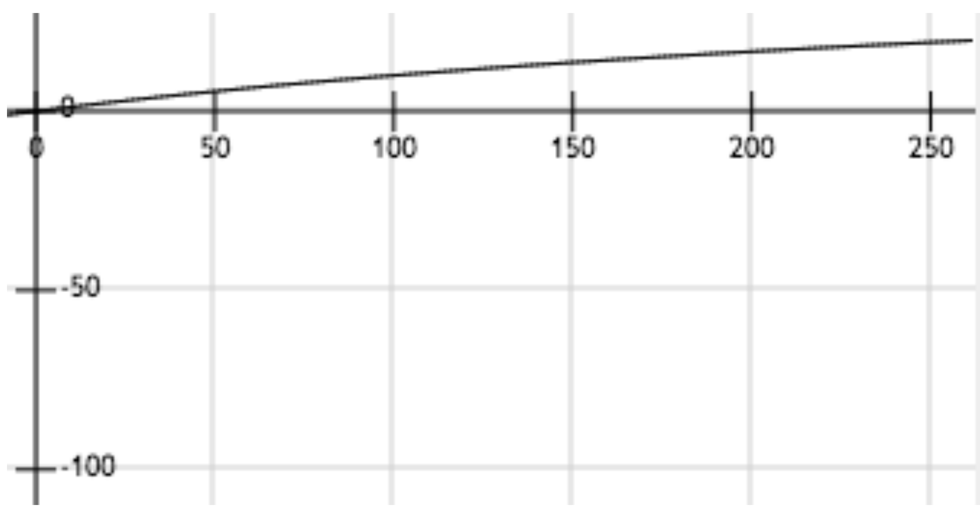
$$TF_k(t,d) = \frac{(k_1 + 1) \times tf(t,d)}{k_1 + tf(t,d)}$$



Pivoted document length normalization

- **Longer documents** will – as a result of containing more terms – have higher tf values **without necessary containing more information**
- needs re-balancing through the use of an inverse function of the document length:

$$TF_P(t,d) = \frac{tf(t,d)}{1 - b + b \times \frac{|d|}{avdl}}$$



Lower-bounding regularization

- There should be a **sufficiently large gap in the score between the presence and absence of a query term** even for very long documents where TF***p*** tends to 0:

$$TF_{\delta}(t,d) = \begin{cases} tf(t,d) + \delta & \text{if } tf(t,d) > 0 \\ 0 & \text{otherwise} \end{cases}$$

BM25 Ranking Function

- Ranking function assuming bag-of-words document representation

$$score(d,q) = \sum_{t \in d \cap q} idf_t \times \frac{tf_{t,d} \cdot (k_1 + 1)}{tf_{t,d} + k_1 \cdot \left(1 - b + b \frac{len_d}{avglen}\right)}$$

  - len_d is the length of document d
  - *avglen* is the average document length in the collection

- Score depends only on query terms

- Values of parameters k_1 and b depend on collection/task
  - k_1 controls term frequency saturation
  - b controls length normalization
  - Default values: $k_1 = 1.2$ and $b = 0.75$

Composition framework

- We apply the various TF normalizations to the raw term frequency (***tf***) through composition.
- For example,
  - TF***p***∘***l*** (= TF***p*** ∘ TF***l***) => $TF_{p \circ l}(t,d) = \frac{1 + \ln[1 + \ln[tf(t,d)]]}{1 - b + b \times \frac{|d|}{avdl}}$
  - TF***k***∘***p*** (= TF***k*** ∘ TF***p***) $= \frac{(k_1 + 1) \times tf(t,d)}{k_1 \times [1 - b + b \times \frac{|d|}{avdl}] + tf(t,d)}$
- The TF components of TF/IDF ad BM25 resp.

Novel composition scheme

- TF$\boldsymbol{l}$∘$\boldsymbol{\delta}$∘$\boldsymbol{p}$ (= TF$\boldsymbol{l}$ ∘ TF$\boldsymbol{\delta}$ ∘ TF$\boldsymbol{p}$) =>

$$TF_{l \circ \delta \circ p}(t,d) = 1 + \ln[1 + \ln[\frac{tf(t,d)}{1 - b + b \times \frac{|d|}{avdl}} + \delta]]$$

- **TF*l*∘δ∘*p*** as it is part of a novel retrieval model
- proved to perform significantly better than Piv+ and BM25+
- in practice on various standard TREC collections
- never been considered before

Experimental setup

- **TREC collections**
- Disks 4&5 (minus the Congressional Record) – 528,155 news releases from early '90s WT10G – 1,692,096 crawled pages from a snapshot of the Web in 1997
  .GOV2 – 25,205,179 crawled pages from various .gov sites in early 2004
- **Evaluation metrics**
- *Precision at 10* (P@10)
  Mean Average Precision (MAP) – only the top-ranked 1000 documents for each run

Experimental setup

- **Evaluation criteria**
- compared weighting models that use the same functions but with a different order of composition (e.g. TF$\boldsymbol{k \circ p}$ vs. TF$\boldsymbol{p \circ k}$) and select the best ones on both metrics.
- The statistical significance of improvement was assessed using the Student's paired t-test considering p-values less than 0.01 to reject the null hypothesis.

Experiments

Table : TF-IDF vs. BM25: an inverse order of composition; bold indicates significant performances

Weighting model	TREC 2004 Robust		TREC9-10 Web		TREC 04-06 Terabyte	
	MAP	P@10	MAP	P@10	MAP	P@10
IDF	0.1396	0.2040	0.0539	0.0729	0.0478	0.0651
TF	0.0480	0.0867	0.0376	0.0833	0.0126	0.0617
TF_p [b=0.20]	0.0596	0.1193	0.0531	0.1021	0.0228	0.0631
TF_p [b=0.75]	0.0640	0.1289	0.0473	0.1000	0.0262	0.0550
TF_l	0.1591	0.3141	0.1329	0.2063	0.1412	0.4215
TF_k	0.1768	0.3269	0.1522	0.2104	0.1569	0.4188
$TF_{p \circ k}$	0.0767	0.1932	0.0465	0.0604	0.0467	0.2081
$TF_{l \circ p}$	0.1645	0.3651	0.0622	0.1854	0.1466	0.4658
$TF_{p \circ l}$	0.1797	0.3647	0.1260	0.1875	0.1853	0.4913
$TF_{k \circ p}$	0.2045	0.3863	0.1702	0.2208	0.2527	0.5342
$TF_{p \circ k} \times$IDF	0.1034	0.2293	0.0507	0.0833	0.0481	0.2168
$TF_{l \circ p} \times$IDF	0.1939	0.3964	0.0750	0.2125	0.1677	0.4919
$TF_{p \circ l} \times$IDF [TF-IDF]	**0.2132**	**0.4064**	**0.1430**	**0.2271**	**0.2068**	**0.4973**
$TF_{k \circ p} \times$IDF [BM25]	**0.2368**	**0.4161**	**0.1870**	**0.2479**	**0.2738**	**0.5383**

Experiments

Table : $\mathrm{TF}_{l \circ \delta \circ p} \times \mathrm{IDF}$ vs. BM25+ and BM25L; bold indicates significant performances

Weighting model	TREC 2004 Robust		TREC9-10 Web		TREC 04-06 Terabyte	
	MAP	P@10	MAP	P@10	MAP	P@10
$\mathrm{TF}_{\delta \circ p \circ k}$	0.1056	0.2349	0.0556	0.0771	0.0531	0.2352
$\mathrm{TF}_{\delta \circ l \circ p}$	0.1807	0.3751	0.0668	0.2021	0.1566	0.4758
$\mathrm{TF}_{l \circ \delta \circ p}$	0.2130	0.4064	0.1907	0.2625	0.2631	0.5597
$\mathrm{TF}_{\delta \circ p \circ l}$	0.2002	0.3876	0.1436	0.2021	0.2055	0.5081
$\mathrm{TF}_{k \circ \delta \circ p}$	0.2155	0.3936	0.1806	0.2292	0.2647	0.5409
$\mathrm{TF}_{\delta \circ k \circ p}$	0.2165	0.3956	0.1835	0.2354	0.2654	0.5369
$\mathrm{TF}_{\delta \circ p \circ k} \times \mathrm{IDF}$	0.1466	0.2723	0.0715	0.1000	0.0660	0.2396
$\mathrm{TF}_{\delta \circ l \circ p} \times \mathrm{IDF}$	0.2096	0.4048	0.0806	0.2292	0.1745	0.4919
$\mathrm{TF}_{l \circ \delta \circ p} \times \mathrm{IDF}$	**0.2495**	**0.4305**	**0.2084**	**0.2771**	**0.2886**	**0.5705**
$\mathrm{TF}_{\delta \circ p \circ l} \times \mathrm{IDF}$ [Piv+]	0.2368	0.4157	0.1643	0.2438	0.2293	0.5047
$\mathrm{TF}_{k \circ \delta \circ p} \times \mathrm{IDF}$ [BM25L]	**0.2472**	**0.4217**	**0.2000**	**0.2563**	**0.2858**	**0.5423**
$\mathrm{TF}_{\delta \circ k \circ p} \times \mathrm{IDF}$ [BM25+]	**0.2466**	**0.4145**	**0.2026**	**0.2521**	**0.2830**	**0.5383**

References

- Composition of TF Normalizations: New Insights on Scoring Functions for Ad Hoc IR, François Rousseau∗ and Michalis Vazirgiannis. SIGIR 2013

Ranked lists comparison

- Typical evaluation setting
  - Set of documents
  - Set of information needs, expressed as queries (typically 50 or more)
  - Relevance assessments specifying for each query the relevant and non-relevant documents

Evaluating ranked results

- Average Precision (AP)
  - average of precision after each relevant document is retrieved
  - Example: AP = 1/1+2/3+3/5+4/7 = 0.7095
  - Mean Average Precision (MAP)
- Precision at K
  - precision after K documents have been retrieved
  - Example: Precision at 10 = 4/10 = 0.4000
- R-Precision
  - For a query with *R* relevant documents, compute precision at R
  - Example: for a query with R=7 relevant docs,
    R-Precision = 4/7 = 0.5714

Rank	Relevant?
1	1
2	0
3	1
4	0
5	1
6	0
7	1
8	0
9	0
10	0

Non-binary relevance

- So far, relevance has been binary
  - a document is either relevant or non-relevant
- The degree to which a document satisfies the user's information need varies
  - Perfect match: *rel = 3*
  - Good match: *rel* = 2
  - Marginal match: *rel = 1*
  - Bad match*:* *rel = 0*

- Evaluate systems assuming that
  - highly relevant documents are more useful than marginally relevant documents, which are more useful than non-relevant ones
  - highly relevant documents are more useful when having higher ranks in the search engine results

Discounted Cumulative Gain

- Cumulative Gain (CG) at rank position *p*

$$CG_p = \sum_{i=1}^{p} rel_i$$

  - Independent of the order of results among the top-*p* positions

- Discounted Cumulative Gain(DCG) at rank position *p*

$$DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2 i}$$

- Normalized Discounted Cumulative Gain (nDCG) at rank position *p*
  - allows comparison of performance across queries
  - compute ideal DCG_p ($IDCG_p$) from perfect ranking of documents in decreasing order of relevance

$$nDCG_p = \frac{DCG_p}{IDCG_p}$$

Outline

- Advanced tf weighting schemes
- Ranked Lists comparison
- **Graph-of-words**

Graph of Word – a novel approach for text mining

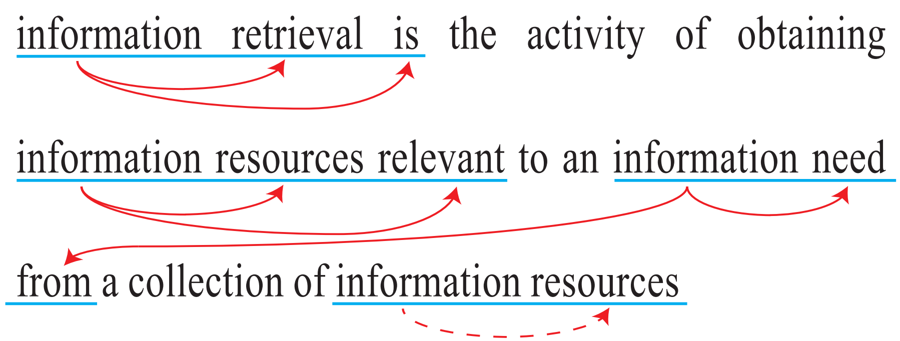
- For efficiency reasons, we used to make the term independence assumption through the bag-of-word representation and the term frequency weighting

- We challenge this assumption by taking into account word dependence and word order through a graph-based representation of a document at indexing time (graph-of-word)

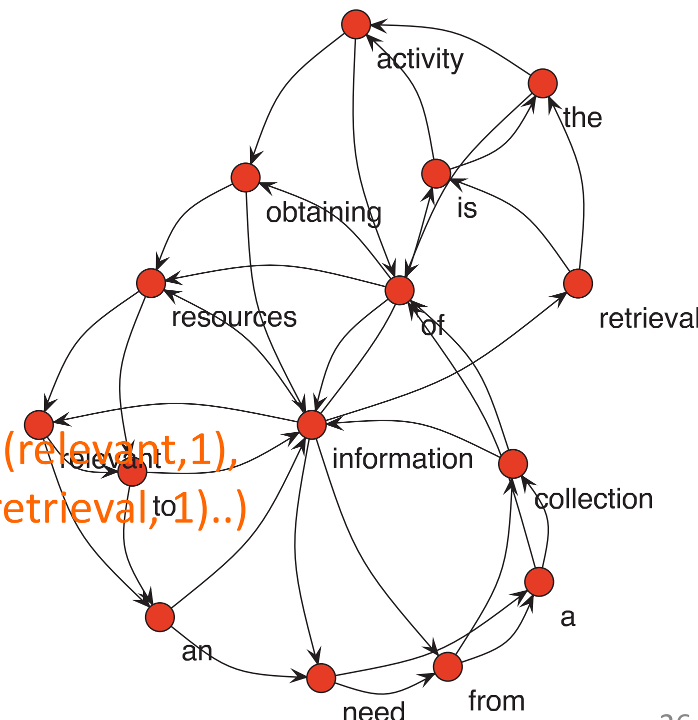
- Graphs have been successfully used in IR to encompass relations between entities and propose meaningful weights (e.g. PageRank[Page et al., 1999])

Graph-based Text Mining

- Instead of the traditional *bag-of-words* (i.e. multiset of terms), we represent a document as a *graph-of-words* to capture *word order* and *word dependency*.



Bag of words: ((activity,1), (collection,1) (information,4), (relevant,1), (resources, 2), (retrieval,1)..)



Graph of Word – a novel approach for text mining

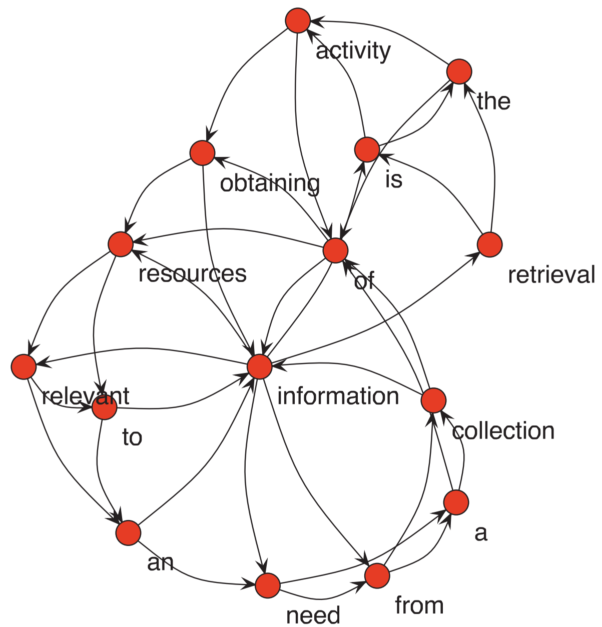
- For efficiency reasons, we used to make the term independence assumption through the bag-of-word representation and the term frequency weighting

- We challenge this assumption by taking into account word dependence and word order through a graph-based representation of a document at indexing time (graph-of-word)

- Graphs have been successfully used in IR to encompass relations between entities and propose meaningful weights (e.g. PageRank [Page et al., 1999])

Graph-based Ad Hoc IR I

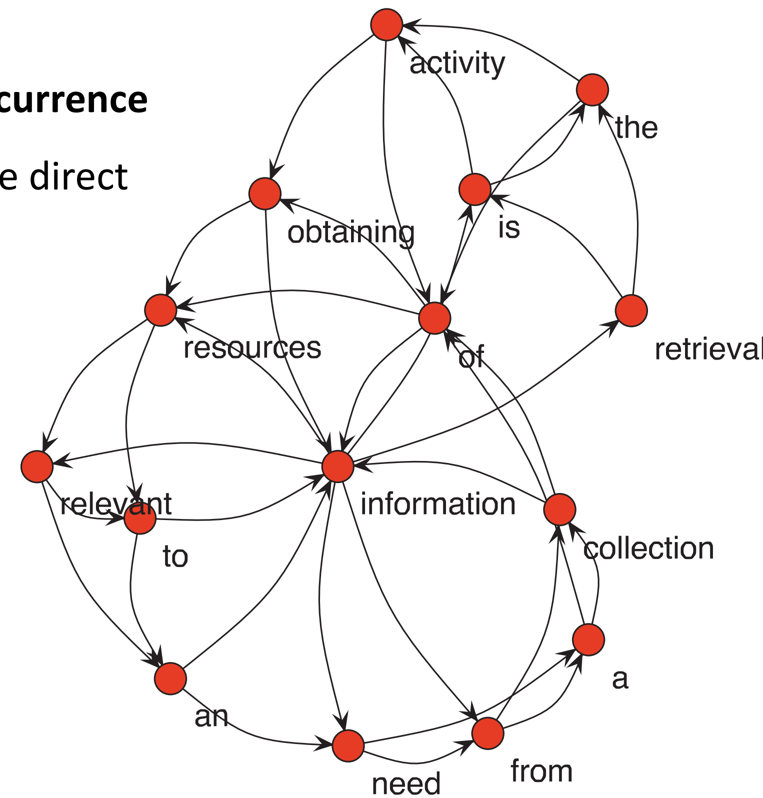
- Ad Hoc Information Retrieval [1,2]
  - Unweighted directed graph
  - The weight of a word in the document: number of neighbors in the graph => favor words that occur with many different other words
  - Robust to varying document length: weight of a word increases only with **new context** of co-occurrences as opposed to the word frequency that increases with any new co-occurrence.



Indegree-BASED TW

- The weight of a term in a document is its **indegree** (numbers of incoming edges) in the **graph-of-word**
- It represents the **number of distinct contexts of occurrence**
- We store the document as a vector of weights in the direct index and similarly in the inverted index
- For example:

information	5
retrieval	1
is	2
the	2
activity	2
of	3
obtaining	2
resources	3
relevant	2
to	2
an	2
need	2
from	2
a	2
collection	2



TF-IDF and BM25

- Term t, document d, collection of size N, term frequency *tf(t, d)*, document frequency *df(t)*, document length |*d*|, average document length *avdl*, asymptotical marginal gain k_1 (1.2), slope parameter *b*

- TF-IDF [Singhal et al., TREC-7]

TF-IDF(t, d) = TF_{pol}-IDF(t, d) = $TF_p o TF_l$(t, d) x IDF(t) = $\left(\frac{1+\log\left(1+\log\left(tf(t,d)\right)\right)}{1-b+b\times\frac{|d|}{avdl}} \right) \times \log\left(\frac{N+1}{df(t)}\right)$

- BM25 [Lv and Zhai, CIKM '11]

BM25(t, d) = $\left(\frac{(k_1+1)\times tf(t,d)}{k_1\times\left(1-b+b\times\frac{|d|}{avdl}\right)+tf(t,d)} \right) \times \log\left(\frac{N+1}{df(t)}\right)$

TW-IDF

- Term t, document *d*, collection of size *N*, term weight *tw(t, d)*, document frequency *df(t)*, document length *|d|*, average document length *avdl*, asymptotical marginal gain k_1 (1.2), slope parameter *b*
- TW-IDF(t, d) = $$\left(\frac{tw(t,d)}{1-b+b\times\frac{|d|}{avdl}} \right) \times \log\left(\frac{N+1}{df(t)}\right)$$
- In the bag-of-word representation, *tw* is usually defined as the term frequency or sometimes just the presence/absence of a term (binary tf)
- In our graph-of-word representation, *tw* is the indegree of the vertex representing the term in the graph

EXPERIMENTS

- DATASETS
- PLATFORM
- EVALUATION
- RESULTS

Datasets

- Disks 1 & 2 (TREC)

  741,856 news articles from Wall Street Journal (1987-1992), Federal Register (1988-1989), Associated Press (1988-1989 and Information from the Computer Select disks (1989-1990)

- Disks 4 & 5 (TREC, minus the Congressional Record)

  528,155 news releases from Federal Register (1994), Financial Times (1991-1994), Foreign Broadcast Information Service (1996) and Los Angeles Times (1989-1990)

- WT10G (TREC)

  1,692,096 crawled pages from a snapshot of the Web in 1997

- .GOV2 (TREC)

  25,205,179 crawled Web pages from .gov sites in early 2004

Datasets (cont.)

Statistic \ Dataset	Disks 1 & 2	Disks 4 & 5	WT10G	.GOV2
# of documents	741,856	528,155	1,692,096	25,205,179
# of unique terms	535,001	520,423	3,135,780	15,324,292
average # of terms (*avdl*)	237	272	398	645
average # of vertices	125	157	165	185
average # of edges	608	734	901	1,185

Table 1: Statistics on the four TREC datasets used; Disks 4&5 excludes the Congressional Record. The average values are computed per document.

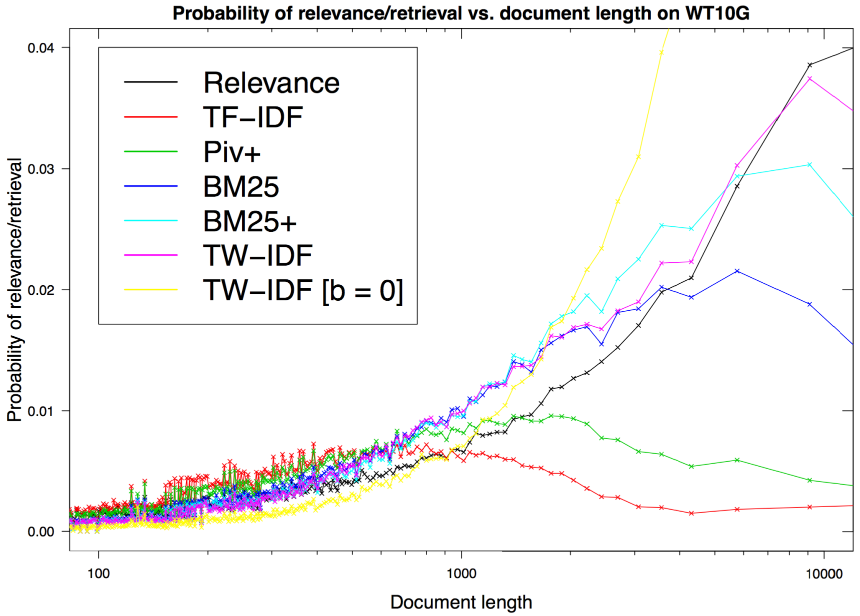
EVALUATION

- Mean Average Precision (MAP) and Precision at 10 (P@10)
  considering only the top-ranked 1000 documents for each run
- Statistical significance of improvement was assessed using the Student's paired t-test
  - R implementation (t.test {stats} package), trec_eval output as input
  - Two-sided p-values less than 0.05 and 0.01 to reject the null hypothesis
- Likelihood of relevance vs. likelihood of retrieval [Singhal et al., SIGIR '96]

- 4 baseline models: TF-IDF, BM25, Piv+ and BM25+
  - Tuned slope parameter *b* for pivoted document length normalization (2-fold cross-validation, odd vs. even topic ids, MAP maximization)
  - Default (1.0) lower-bounding gap ∂ [Lv and Zhai, CIKM '11]

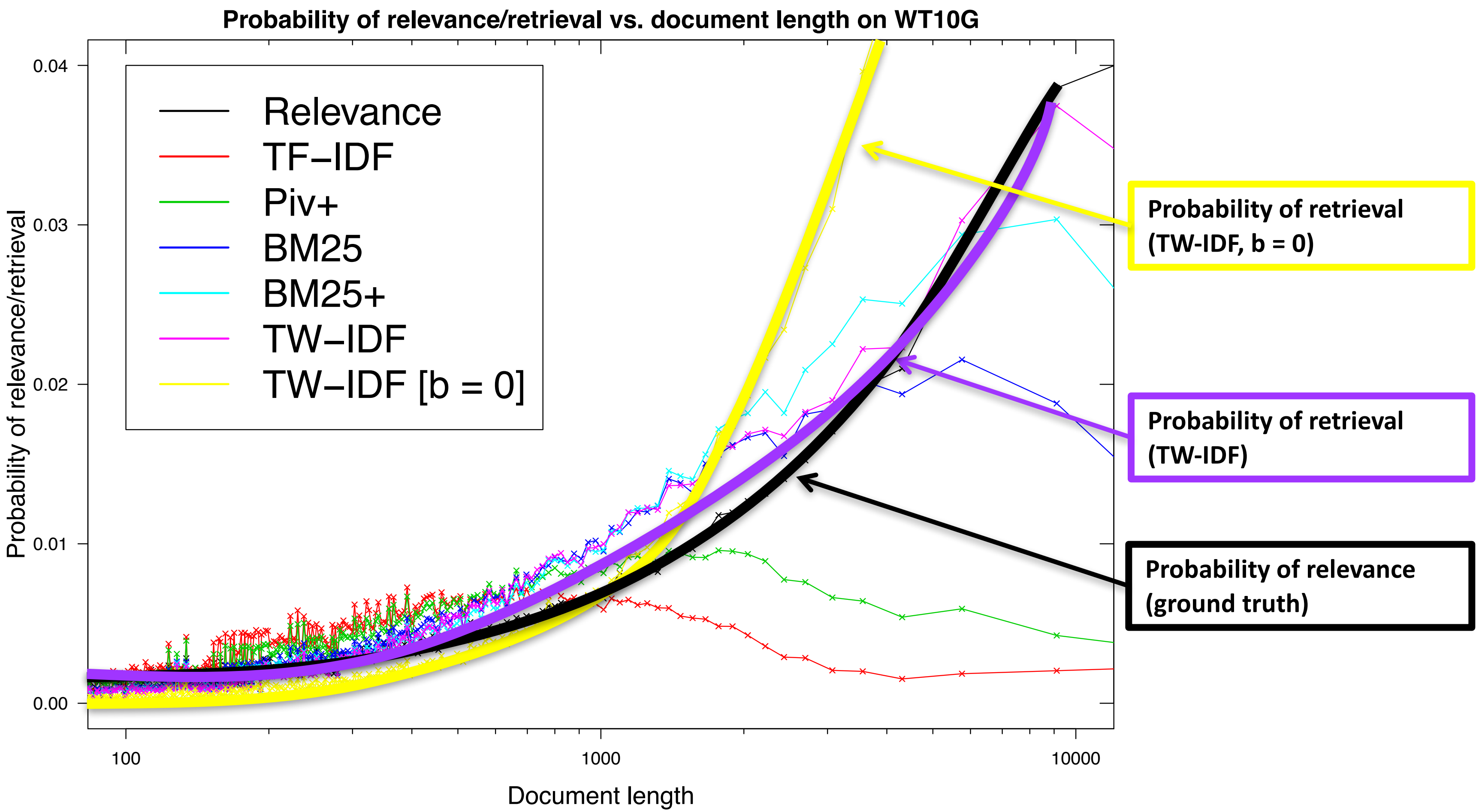
Graph-based Ad Hoc IR II

- Evaluation in terms of:
  - Mean Average Precision
  - Precision@10
  - Probability of relevance vs. probability of retrieval



Model	b	TREC1-3 Ad Hoc		TREC 2004 Robust		TREC9-10 Web		TREC 2004-2006 Terabyte	
		MAP	P@10	MAP	P@10	MAP	P@10	MAP	P@10
TF_{pol}	0.20	0.1471	0.3960	0.1797	0.3647	0.1260	0.1875	0.1853	0.4913
TF_{kop}	0.75	0.1346	0.3533	0.2045	0.3863	0.1702	0.2208	0.2527	0.5342
TW	none	0.1502	0.3662	0.1809	0.3273	0.1430	0.1979	0.2081	0.5021
TW_p	0.003	**0.1576****	**0.4040****	**0.2190****	**0.4133****	**0.1946****	**0.2479****	**0.2828****	**0.5407****
TF-IDF	0.20	0.1832	0.4107	0.2132	0.4064	0.1430	0.2271	0.2068	0.4973
BM25	0.75	0.1660	0.3700	0.2368	0.4161	0.1870	0.2479	0.2738	0.5383
TW-IDF	0.003	**0.1973****	**0.4148***	**0.2403****	**0.4180***	**0.2125****	**0.2917****	**0.3063****	**0.5633****

Likelihood of relevance vs. likelihood of retrieval



GoW for keyword extraction for documents

Keywords are used everywhere:

- looking up information on the Web (e. g., via a search engine bar)
- finding similar posts on a blog (e. g., tag cloud)
- for ads matching (e. g., AdWords' keyword planner)
- for research paper indexing and retrieval (e. g., SpringerLink)
- for research paper reviewer assignment (e. g., ECIR '15!)

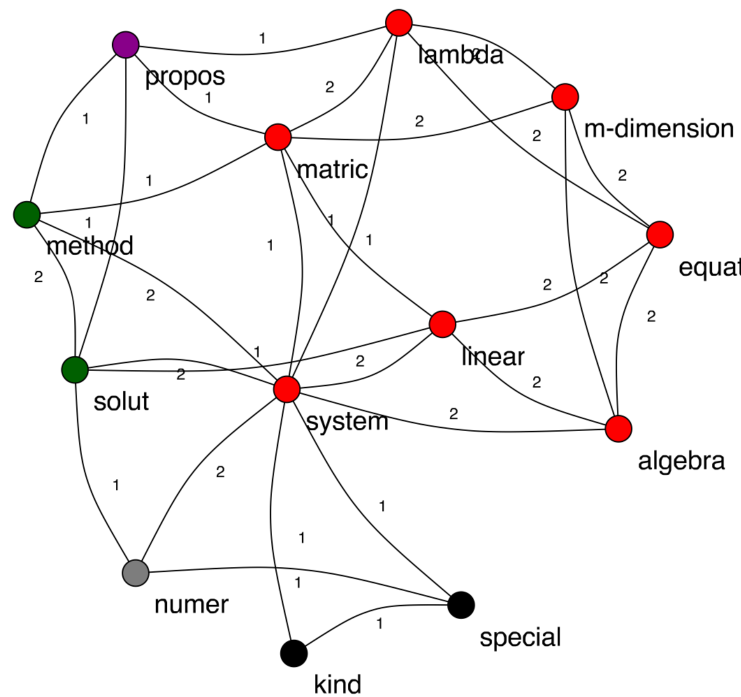
Applications are numerous:

- **summarization** (to get a gist of the content of a document)
- **information filtering** (to select specific documents of interest)
- **indexing** (to answer keyword-based queries)
- **query expansion** (using additional keywords from top results)

Graph-based Keyword Extraction

- Single-Document Keyword Extraction [3]
  - Elect the most cohesive sets of words in the graph as keywords
  - Use k-core decomposition to extract the main core of the graph
  - Weighted edges as opposed to Ad Hoc IR (single-document => no normalization)

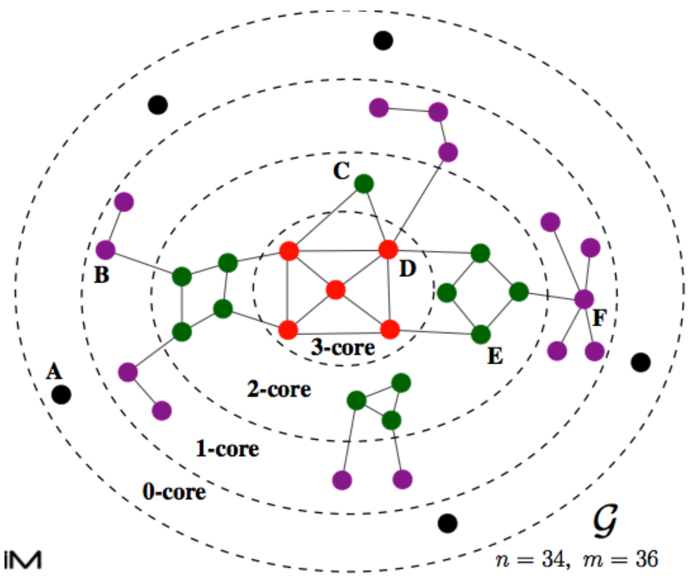
A method for solution of systems of linear algebraic equations with m-dimensional lambda matrices.
A system of linear algebraic equations with m-dimensional lambda matrices is considered. The proposed method of searching for the solution of this system lies in reducing it to a numerical system of a special kind.



Keywords manually assigned by human annotators
linear algebra equat; numer system; m-dimension lambda matric

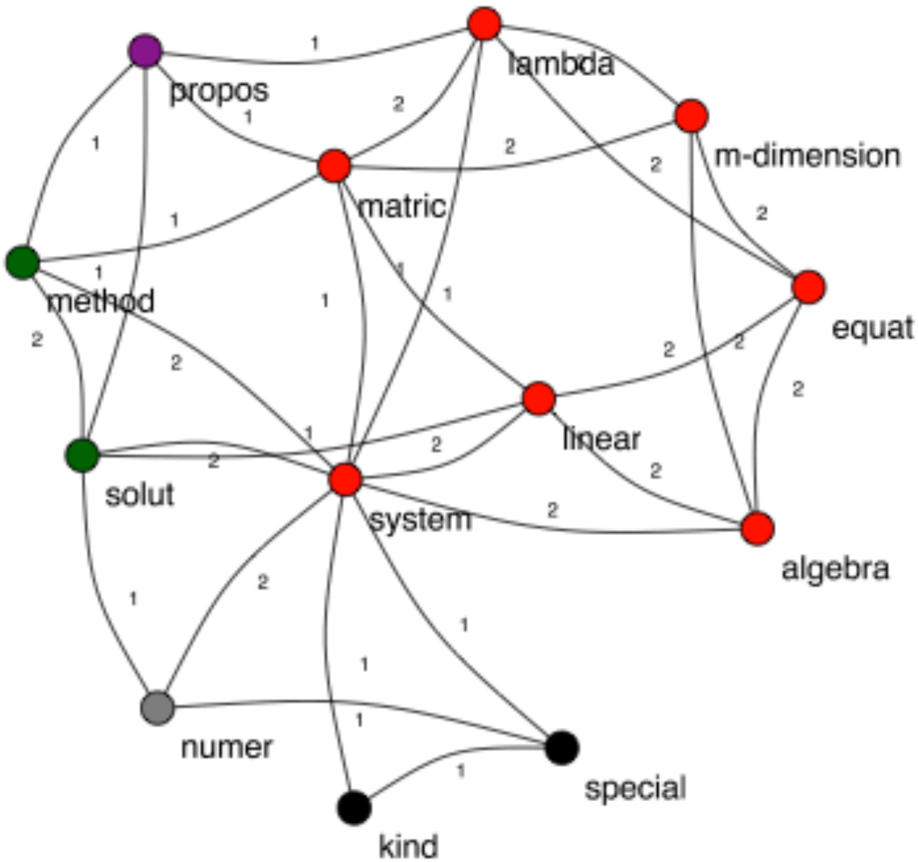
Graph-based keyword extraction

Two existing graph-based keyword extractors: PageRank [Mihalcea and Tarau, 2004] HITS [Litvak and Last, 2008]
Two iterative graph algorithms that assign an influence score to a node based on its centrality (top ones supposedly the most representative).
⇒ we propose to retain the main core of the graph to extract the nodes based on their centrality and cohesiveness.



K-core decomposition of the graph

Pagerank vs. main core

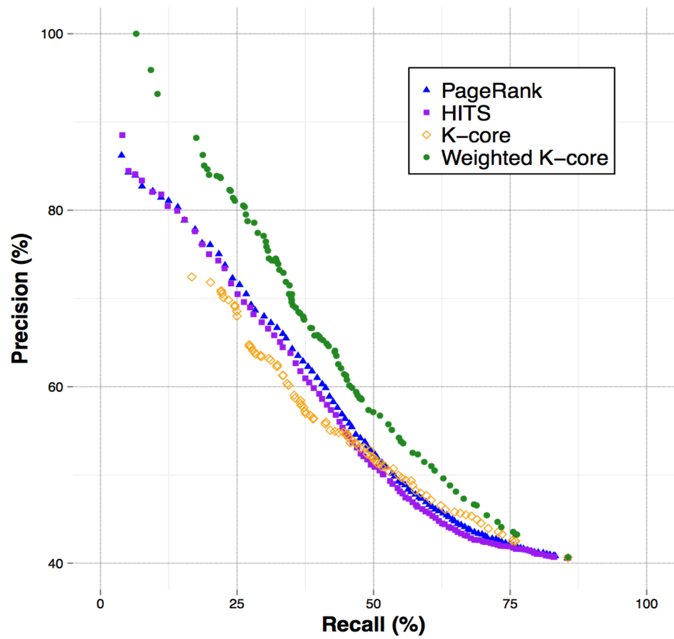


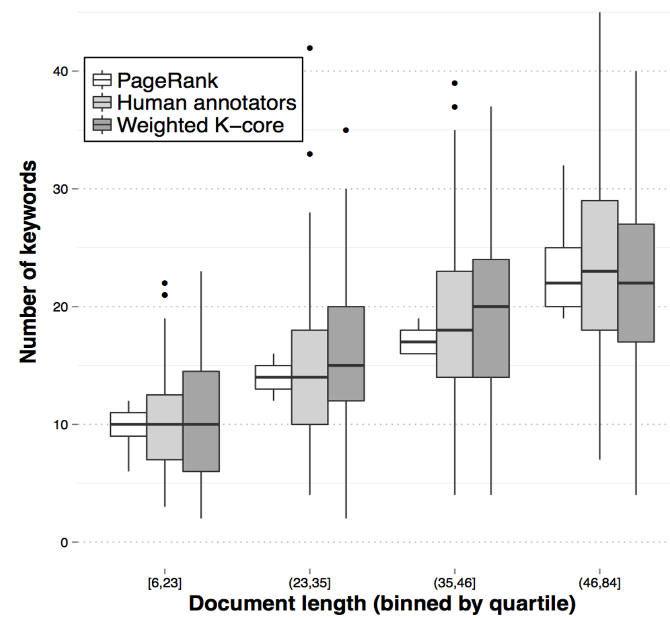
Keywords manually assigned by human annotators
linear algebra equat; numer system; m-dimension lambda matric

WK-core		PageRank	
system	6	**system**	1.93
matric	6	**matric**	1.27
lambda	6	solut	1.10
linear	6	**lambda**	1.08
equat	6	**linear**	1.08
algebra	6	**equat**	0.90
m-dim...	6	**algebra**	0.90
method	5	**m-dim...**	0.90
solut	5	propos	0.89
propos	4	method	0.88
numer	3	special	0.78
specia	2	**numer**	0.74
kind	2	kind	0.55

Graph-based Keyword Extraction II

- Evaluation in terms of:
  - Precision
  - Recall
  - F1-score
  - Precision/recall
  - # of keywords





Graph	Dataset	Macro-averaged precision (%)				Macro-averaged recall (%)				Macro-averaged F1-score (%)			
		PageRank	HITS	K-core	WK-core	PageRank	HITS	K-core	WK-core	PageRank	HITS	K-core	WK-core
undirected edges	Hulth2003	58.94	57.86	46.52	**61.24***	42.19	41.80	**62.51***	50.32*	47.32	46.62	49.06*	**51.92***
	Krapi2009	50.23	49.47	40.46	**53.47***	48.78	47.85	**78.36***	50.21	49.59	47.96	46.61	**50.77***
forward edges	Hulth2003	55.80	54.75	42.45	**56.99***	41.98	40.43	**72.87***	46.93*	45.70	45.03	**51.65***	50.59*
	Krapi2009	47.78	47.03	39.82	**52.19***	44.91	44.19	**79.06***	45.67	45.72	44.95	46.03	**47.01***
backward edges	Hulth2003	59.27	56.41	40.89	**60.24***	42.67	40.66	**70.57***	49.91*	47.57	45.37	45.20	**50.03***
	Krapi2009	51.43	49.11	39.17	**52.14***	49.96	47.00	**77.60***	50.16	**50.51**	47.38	46.93	50.42

References

- Main core retention on graph-of-words for single-document keyword extractionF Rousseau, M Vazirgiannis, European Conference on Information Retrieval, 382-393
- Graph-of-word and TW-IDF: new approach to ad hoc IRF Rousseau, M Vazirgiannis Proceedings of the 22nd ACM CIKM