Graph-of-word and TW-IDF: New Approach to Ad Hoc IR

François Rousseau[*], Michalis Vazirgiannis[*†‡]
[*]LIX, École Polytechnique, France
[†]Department of Informatics, AUEB, Greece
[‡]Institut Mines-Télécom, Télécom ParisTech, France
rousseau@lix.polytechnique.fr, mvazirg@aueb.gr

ABSTRACT

In this paper, we introduce novel document representation (graph-of-word) and retrieval model (TW-IDF) for ad hoc IR. Questioning the term independence assumption behind the traditional bag-of-word model, we propose a different representation of a document that captures the relationships between the terms using an unweighted directed graph of terms. From this graph, we extract at indexing time meaningful term weights (TW) that replace traditional term frequencies (TF) and from which we define a novel scoring function, namely TW-IDF, by analogy with TF-IDF. This approach leads to a retrieval model that consistently and significantly outperforms BM25 and in some cases its extension BM25+ on various standard TREC datasets. In particular, experiments show that counting the number of different contexts in which a term occurs inside a document is more effective and relevant to search than considering an overall concave term frequency in the context of ad hoc IR.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval models; H.3.1 [**Content Analysis and Indexing**]: Linguistic processing

General Terms

Theory, Algorithms, Experimentation

Keywords

IR theory; scoring functions; graph representation of document; graph-of-word; graph-based term weighting; TW-IDF

1. INTRODUCTION

In this paper, we introduce novel graph-based document representation and scoring function for ad hoc Information Retrieval (IR). For the past decades, there has been a lot of effort put in research in IR, popularized by Web Search and its extensive use to find everyday information by most people. Because the users expect to find the most relevant documents first with regard to their query, an important issue in IR has been the design of a function that relatively scores documents with regard to this query. Most of the existing state-of-the-art techniques rely on the so-called *bag-of-word* representation of a document computed at *indexing time*. This corresponds to an unordered set of pairs of terms and term frequencies (TF); the underlying assumption being that each term is considered independent of one another. The differences between the techniques lie in the way the relevance of a document for a given query is assessed at *query time*, i.e. the definition of the retrieval model (also called scoring function or weighting model). There exist several categories of methods, in particular *vector space model* (TF-IDF [30]), *probabilistic* (BM25 [25]) and *language modeling* (Dirichlet prior [33]) approaches and the *divergence from randomness* framework (PL2 [1]).

In our work, we intended to go one step further and we propose a different document representation, one that captures the relationships between the terms, questioning the *term independence assumption.* The impact of the term order has been a popular issue in text mining and relationships between the terms in general is claimed to play an important role in text processing [32]. So far, taking it into account has been more expensive without being significantly more effective and thus less used in IR in practice; the main reason being a lack of an efficient representation. Here, we propose a novel document representation based on an unweighted directed graph whose vertices represent terms, whose edges represent co-occurrences between the terms within a fixed-size sliding window and whose edge direction represents term order. We claim that the use of a graph saves more useful information than a standard vector while introducing only a small overhead in terms of computational costs (both in time and space) at indexing time, hence no delay to the IR system's response time to the user at query time. The proposed graph-of-word model brings a lot of new possibilities and applications to IR and we chose document scoring functions as a first application and as a way to evaluate our approach compared to the traditional one. The design of an effective retrieval model is a cornerstone issue in IR and it seemed natural to tackle this problem first.

The rest of the paper is organized as follows. Section 2 overviews related work in terms of graph representations of text, term weighting strategies and scoring function design in ad hoc IR. Section 3 explains the motivation behind this novel approach and why we think the use of graph is essential for better ad hoc IR. Section 4 presents the adopted graph-

CIKM'13, Oct. 27–Nov. 1, 2013, San Francisco, CA, USA.
Copyright 2013 ACM 978-1-4503-2263-8/13/10 ...$15.00.
http://dx.doi.org/10.1145/2505515.2505671.

of-word model and the associated term weighting strategy. Section 5 introduces the proposed novel scoring function, denoted TW-IDF, and its relation with TF-IDF and BM25. Section 6 describes the results we obtained over four standard TREC datasets (two of news articles and two of Web pages) and shows how TW-IDF performs significantly better than TF-IDF and BM25 and in some cases their extensions. Section 7 discusses and interprets these performances. Finally, section 8 concludes and mentions future applications for Graph-based Information Retrieval.

2. RELATED WORK

Graph representations of text and scoring function design for ad hoc IR are two research topics widely explored on their own but little work has been done around graph-based IR in terms of document representation and weighting model altogether. Blanco and Lioma in [2, 3] are to the best of our knowledge the only works closely related to ours. Nevertheless, their graph definition differs from ours (see subsection 4.1), they focused on the size of the sliding window of co-occurrences (subsection 4.4) and they applied random walk for term weighting (subsection 4.5). This is not the approach we followed: we preferred an indegree-based term weight to replace the traditional term frequency. And we rather concentrated on the scoring function and the satisfaction of a set of heuristic retrieval constraints defined in [6, 14], in particular concavity, document length normalization and lower-bounding regularization (subsection 5.3).

Text can be represented as a graph in various ways. We refer to [3] for an in-depth review of all the graph representations of a document in the relevant literature. We note in particular the works of Erkan and Radev in [5] and Mihalcea and Tarau in [16] that both proposed systems, namely LexRank and TextRank, that use centrality measures to extract the most salient vertices of a graph, corresponding to sentences in their case. Those systems and their extensions have had applications in text summarization [5, 16], keyword detection [13] and word sense disambiguation [17] among others. What a vertex of a graph represents depends entirely on the context of use and on the level of granularity needed. This can be a sentence [5, 16], a word [16, 3] or even a character [9]. Similarly, an edge represents a meaningful relation between two vertices, either linguistic (e.g. syntactic [8] or semantic [18]) or statistical (e.g. co-occurrence [7]) depending on the use case. Edges can be directed (e.g. according to Jespersen's Rank Theory [10] or POS tags [3]) or undirected [5, 16], weighted (e.g. frequency [13] or strength [5, 16] of the relation) or unweighted [3].

Designing and optimizing retrieval models are cornerstone issues in IR because a novel or an improved retrieval model would lead to improved performance for all search engines. There have been many works done on both aspects. We refer to [15] for a review of the different categories of methods that have been proposed. In particular, we highlight *vector space model* (TF-IDF [30]), *probabilistic* (BM25 [25]) and *language modeling* (Dirichlet prior [33]) approaches and the *divergence from randomness* framework (PL2 [1]). Regarding improvements that benefit to all these scoring functions, Fang *et al.* introduced in [6] heuristic retrieval constraints that any weighting model should satisfy. Lv and Zhai recently proposed in [14] an additional lower-bounding regularization that initial models failed to include. When proposing a novel scoring function such as TW-IDF in our case, one has to take into account these mathematical properties in order to build a more robust model.

Note that all the retrieval methods we mentioned rely on a bag-of-word representation of a document and a frequency-based term weighting. They all assume a vector of term frequencies (TF) and a vector of document frequencies (DF) for each term. More generally, a retrieval model could be defined as a function of a term weight (TW) and a document weight (DW). In this context, there have been some alternatives proposed to challenge the traditional frequency-based TW (e.g. graph-based TW [2, 3] or POS-based TW [12]). But overall less effort has been devoted to proposing a different term weighting strategy compared to a novel retrieval model still based on a bag-of-word representation and term frequencies. In our work, we propose novel document representation (graph-of-word), term weighting (indegree) and scoring function (TW-IDF).

3. MOTIVATION

Semantically speaking, word order and word dependence do matter. *Mary is quicker than John* and *John is quicker than Mary* are clearly different phrases. Yet, their bag-of-word representation is the same. Still, it seems intuitive that two documents with similar bag-of-word representations are similar in content [15] if not in meaning and thus, should be both retrieved for a query on that content. Moreover, this has shown to work already well in practice for ad hoc IR, whether it be the *term frequency vector* representation in the vector space model [27] or be it the *term independence assumption* in probabilistic IR [24] or be it the *unigram language model* in language modeling IR [22]. Nevertheless, the impact of the term order has been a popular issue and relationships between the terms in general is claimed to play an important role in text processing [32]. This motivated us to find a representation that would capture these relationships while being as efficient as the traditional one at query time. Indeed, effectiveness should not be improved at the cost of efficiency. The users are expecting both instant and relevant results and are not ready to sacrifice one for the other, at least in the context of Web search.

Graphs have already been successfully used in Search and Information Retrieval. Page *et al.* with PageRank [21] is probably one of the most seminal works that applies link analysis to the Web's structure. The algorithm computes the probability of the user ending at a given Web page and provides at indexing time meaningful static document scores to boost the dynamic scores computed from the scoring function at query time. In fact, Blanco and Lioma in [2, 3] apply PageRank on their graph-of-word to get static term weights as an alternative to the term frequencies, favoring terms that appear in a lot of different contexts. More recently, Bonchi *et al.* proposed in [4] a novel method for query recommendation based on the computation of the center-piece subgraph for a given query in a query-flow graph. Informally, it consists of a small subgraph that best captures the connections between the source nodes (here, the query terms) and whose vertices correspond to the queries to be recommended from the original query.

Two examples of works that use graph representation to encompass relations between entities, either Web pages or Web queries, and to compute entity score through either vertex's weight or subgraph. This led us to explore graph-based document representation and its use in ad hoc IR as an alternative to the traditional techniques that pre-suppose independence of the terms. In the next section, we will present our novel graph-of-word model and its graph term weights as opposed to the classic bag-of-word model and its term frequencies.

4. FRAMEWORK

4.1 Graph-of-word

We represent a textual document (typically a Web page) as a *graph-of-word* that corresponds to an *unweighted directed graph* whose vertices represent unique terms, whose edges represent co-occurrences between the terms within a fixed-size sliding window and whose edge direction represents term order. We prefer to use *term* rather than *word* because tokenization and feature selection algorithms (such as stopword removal) have already been applied if needed. One could then argue that it is in fact a graph-of-term but we chose this denomination by analogy with the bag-of-word, which is in practice a bag-of-term as well. The underlying assumption is that all the words present in a document have some relationships with the others, modulo a window size, outside of which the relationship is not taken into consideration. This is a statistical approach as it links all co-occurring terms without considering their meaning or function in the text. We shall discuss in the following subsections the choices made in terms of edge direction, edge weight and window size but we will first show how it works in practice.

An example of graph creation is given in Figure 1. The source text is an extract of Wikipedia's definition of IR: "Information retrieval is the activity of obtaining information resources relevant to an information need from a collection of information resources"[1]. The text is tokenized, lowercased and parsed. An edge (red arrow) is drawn between a term and the following two (window size set to 3 in this example, red line). A solid arrow represents a new directed edge while a dashed arrow an already existing one in the graph. We only show the edges created from the vertex corresponding to "information" for clarity purposes.

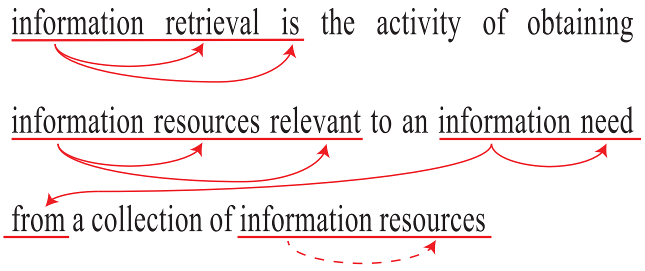
information retrieval is the activity of obtaining
information resources relevant to an information need
from a collection of information resources

Figure 1: Example of graph creation from text parsing. Solid arrows represent new directed edges while dashed arrows already existing ones in the graph. The sliding windows in red are of size 3.

Figure 2 corresponds to the resulting unweighted directed graph where each vertex represents a unique term and each edge a co-occurrence of the two terms in at least one window of size 3. The graph visualization has been automatically generated using the Kamada-Kawai force-based algorithm [11] and the JUNG library [19].

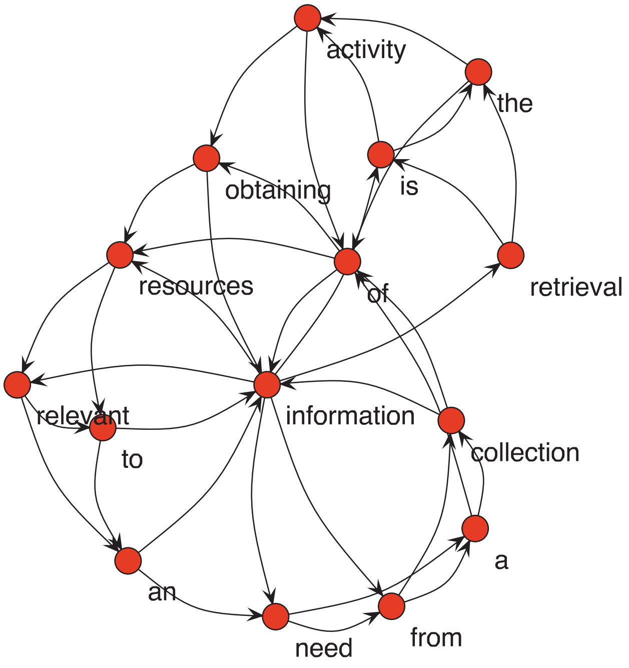

Figure 2: Example of graph representation of a text: an unweighted directed graph in which an edge indicates at least one directed co-occurrence of the two terms in a window of size 3 in the text.

4.2 Directed vs. undirected graph

Initially, we considered undirected edges following related works [5, 16, 2, 3] because an edge captures a co-occurrence of two terms whatever the respective order between them is. Later, we explored edge direction as a way of capturing the natural flow of a text (from left to right, at least in English) as illustrated in Figure 1. This seemed like the intuitive way to encompass term order in the graph-of-word model and yields in practice to better results.

Even though the gain was statistically significant (two-sided p-value less than 0.05 using the Student's t-test) compared to an undirected graph, the increase in MAP and P@10 was still small (less than a 1%) and the impact of the term order on the graph term weight for ad hoc IR has yet to be proved essential. For other applications like phrasal indexing or document summarization, it may be of more importance. Note that there are graph representations of text that direct the edges according to a grammatical hierarchy such as POS-tagging (e.g. adjectives pointing to nouns pointing to verbs) but those approaches proved empirically to be more expensive (e.g. use of an NLP parser such as TreeTagger[2]) while not more effective.

4.3 Weighted vs. unweighted graph

Initially, we considered both unweighted and weighted edges in the graph. It seemed natural to weight the edges by the number of co-occurrences of the two terms in the text. For instance, in Figure 2, the edge from "information" to "resource" could have a weight of 2 to indicate that this context occurs more often (twice in the original text) and thus contains more information a priori. We even tested with

[1] http://en.wikipedia.org/wiki/Information_retrieval

[2] http://www.cis.uni-muenchen.de/~schmid/.../TreeTagger

boosted weights that depend on the distance between the two terms within the considered sliding window. Indeed, in Figure 1, the additional weight of the association "information – resources" should be a priori more important than the one for "information – relevant" in the window "information resources relevant". The closer two terms are, the stronger their relationship should be.

However, in practice, the use of unweighted edges consistently led to better results and that is why we present our graph-of-word as an unweighted graph. We will propose a possible explanation in subsection 7.2 when discussing future work and extensions of our current model.

4.4 Fixed vs. parameterized window size

Blanco and Lioma in [2, 3] considered the window size to be yet another parameter of their model. Following their recommendations, the size should range from 6 to 30. In practice, we did not observe a significant improvement between the various values and we set it to a default value of 4. This is less than in their experiments because we are using stopword removal as a preprocessing step and they did not. Hence, this reduces the range of impact of term dependence. We also chose the minimum value for which it is significant since the computational cost in time of the graph creation is linearly proportional to the window size.

Note that the sliding window is moved along the entire document and not at a smaller level like a sentence or a phrase. We did explore this approach using the OpenNLP Sentence Detector[3] but it did not yield to better results with the cost of using this extra pre-processing on the text.

4.5 Degree-based term weighting

Most of the existing scoring functions such as TF-IDF and BM25 rely on a bag-of word representation of a document and thus, use the term frequency to compute a document score. More generally, as mentioned in section 2, we can define a retrieval model as a function based on a term weight (TW) rather than restricting it to a term frequency (TF). Similarly to the weighting models of the classic IR, we can then define TW in the context of a graph-of-word instead of a bag-of-word. It is simply the weight of the vertex corresponding to that particular term instead of its frequency.

Unlike [2], we chose to keep the vertex weight definition simple – its degree – and more precisely the indegree for a directed graph. Yet we did evaluate other functions but we will not report the results in section 6 since they take more time to compute and did not yield to better results in practice. These functions fall into four categories: measures based on the *degree* (connections between a vertex and its neighbors), *closeness* (lengths of the shortest paths between a vertex and the other vertices), *betweenness* (frequency of the vertex on the shortest paths between other vertices) and *eigenvectors* (spectral decomposition of the adjacency matrix). Note that measures such as PageRank, that belongs to the last category, assign a floating-point weight rather than a small integer and that requires some quantization for efficient compression and storage while our degree-based model does not suffer from this shortcoming. This is another reason for us to keep the vertex weight definition simple. It is actually the reason why in classic IR people store the raw term frequency in the inverted index rather than the normalized version used at query time even if this means computing it every time. There was no mention of this issue in [2, 3] but we think it does matter. Moreover, the use of PageRank weights did not lead to better results than the degree-based ones.

4.6 Term weight indexing

Following PageRank's idea of computing vertices' scores from a graph and storing only those (not the graph structure itself), we only index the term weights and actually replace the term frequencies by them in the inverted index. This allows efficient retrieval at query time. One could argue that we lose some information in the process and it is probably true for other applications such as document summarization or query expansion. But, for the scoring function itself, we only need a term weight that contains more information than the raw term frequency. Since TW is based on the indegree, it already embeds the relationships we wanted to capture with the graph just like PageRank's weights embed the contributions of the other Web pages. It is not excluded that in the future we store the graphs but this is another challenge beyond the scope of the current paper as today's collections consist of millions to billions of documents, that many graphs to index and retrieve efficiently.

5. SCORING FUNCTION

We designed our novel scoring function, denoted TW-IDF, by analogy with TF-IDF and BM25. We considered the successive TF normalizations historically applied on raw term frequencies and we used them in the context of graph term weights. We assumed that TW embeds more information than TF while needing the same regularizations.

5.1 Notations

Fang *et al.* introduced in [6] and later Lv and Zhai in [14] a set of heuristic retrieval constraints for scoring functions in ad hoc IR. In order to satisfy those constraints, one successively applies a set of mathematical functions (TF normalizations) on the raw term frequency (tf). In particular, a *concave function* (TF$_k$ or TF$_l$), a *pivoted document length normalization* (TF$_p$) and a *lower-bounding regularization* (TF$_\delta$). Briefly, the use of a concave function allows us to decrease the marginal gain of seeing an additional occurrence of a term inside the same document. Indeed, the change in the score caused by increasing tf from 1 to 2 should be much larger than the one caused by increasing tf from 100 to 101. The pivoted document length normalization takes care of documents of varying length by normalizing by the document length (term frequencies rather than counts) and by pivoting so that the probability of retrieval for longer documents still matches their probability of relevance [29]. Finally, the lower-bounding regularization ensures the existence of a sufficiently large gap in the score between the presence and absence of a query term even for very long documents where the overall TF component tends to 0, compensating the potential null limit introduced by TF$_p$ [14]. These functions are defined as follows:

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

$$TF_l(t,d) = 1+\ln[1+\ln[tf(t,d)]] \tag{2}$$

$$TF_p(t,d) = \frac{tf(t,d)}{1-b+b\times\frac{|d|}{avdl}} \tag{3}$$

[3] http://opennlp.apache.org/.../opennlp.html

$$TF_{\delta}(t,d) = \begin{cases} tf(t,d) + \delta & \text{if } tf(t,d) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where $tf(t,d)$ is the term frequency of the term t in the document d, k_1 is a constant set by default to 1.2 corresponding to the asymptotical maximal gain achievable by multiple occurrences compared to a single occurrence, $b \in [0,1]$ the slope parameter of the tilting, $|d|$ the document length, $avdl$ the average document length across the collection of documents and δ the lower-bounding gap set by default to 1.0.

The TF component of a TF×IDF weighting model is then built by composing the functions successively, satisfying overall all the heuristic retrieval constraints as presented in [26]. The order of composition differs between retrieval models. We will denote hereinafter by TF_{pol} ($=\text{TF}_p \circ \text{TF}_l$) a model that applies the log-based concavity first (TF_l) and then the pivoted document length normalization (TF_p) and similarly for other compositions.

5.2 TF-IDF and BM25

By TF-IDF, we refer to the TF×IDF weighting model defined in [30], often called *pivoted normalization weighting*. The scoring function corresponds to $\text{TF}_{pol}\times\text{IDF}$:

$$TF\text{-}IDF(t,d) = \frac{1 + \ln[1 + \ln[tf(t,d)]]}{1 - b + b \times \frac{|d|}{avdl}} \times \log \frac{N+1}{df(t)} \quad (5)$$

where b is set by default to 0.20, $df(t)$ is the document frequency of term t across the collection and N the size of the collection.

By BM25, we refer to the scoring function defined in [25], often called *Okapi weighting*. It corresponds to $\text{TF}_{kop}\times\text{IDF}$ when we omit the query frequency normalization. The within-document scoring function of BM25 is written as follows when using the IDF formula from TF-IDF to avoid negative values, following [6]:

$$BM25(t,d) = \frac{(k_1 + 1) \times tf(t,d)}{K + tf(t,d)} \times \log \frac{N+1}{df(t)} \quad (6)$$

where $K = k_1 \times (1 - b + b \times \frac{|d|}{avdl})$ and b set to 0.75.

We will also compare our novel approach to extensions of these two models that take into account the lower-bounding regularization (TF_δ) introduced in [14], namely Piv+ (that corresponds to $\text{TF}_{\delta \circ pol}\times\text{IDF}$) and BM25+ ($\text{TF}_{\delta \circ k \circ p}\times\text{IDF}$).

5.3 TW-IDF

By TW-IDF, we refer to our final model $\text{TW}_p\times\text{IDF}$ (TW_p is defined by analogy with TF_p):

$$TW\text{-}IDF(t,d) = \frac{tw(t,d)}{1 - b + b \times \frac{|d|}{avdl}} \times \log \frac{N+1}{df(t)} \quad (7)$$

where $tw(t,d)$ is the weight of the vertex associated with the term t in the graph-of-word representation of the document d. In the experiments of section 6, the weight is the indegree. b is set to 0.003 and does not require tuning. This constant value consistently produced good results across various collections. Its two orders of magnitude less than TF-IDF (0.20) and BM25 (0.75) can be explained by the structure of the graph-of-word. Since there is no weight on the edges, the indegree of a vertex does not increase linearly with the document length like the term frequency does: it is incremented only when a new context of occurrence appears. Thus, this requires less tilting (but still some pivoting as observe empirically, see subsection 6.4 of the experiments section).

Note that the model does not explicitly include a concave normalization (TW_k or TW_l). In fact, applying such a function worsens sometimes the results. This can be explained by the use of unweighted edges in the graph representation. Recall that the raw term frequency was historically dampened so that the difference between 1 and 2 and 100 and 101 in terms of tf values is much more important for the former. Here, in the context of a graph-of-word, an additional edge is added to the graph only if the context of occurrence for the term is new. This holds the same amount of information whatever the tw value already is. Hence, the absence of such regularization for effectiveness reasons.

Similarly, there is no explicit lower-bounding regularization as well. This can be explained by the two orders of magnitude less for b. In the original paper [14], it was noted that the pivoted document length normalization could yield to null scores for very long documents (as $|d|$ becomes much larger than $|avdl|$). Here, this would happen in TW-IDF for documents a hundred times longer than for TF-IDF or BM25 and there is none in today's collections. Anyway, if this were the case, then it would only require an additional composition with the function TW_δ that we omit for now in the implementation for efficiency reasons.

6. EXPERIMENTS

6.1 Datasets and evaluation

We used four standard TREC collections of documents to carry out our experiments: Disks 1&2, Disks 4&5 (minus the Congressional Record), WT10G and .GOV2. Disks 1&2 includes 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)[4]. Disks 4&5 contains 528,155 news releases from Federal Register (1994), Financial Times (1991-1994), Foreign Broadcast Information Service (1996) and Los Angeles Times (1989-1990)[4]. WT10G consists of 1,692,096 crawled pages from a snapshot of the Web in 1997. .GOV2 corresponds to a crawl of 25,205,179 .gov sites in early 2004. We present in Table 1 some basic statistics on the datasets. The document length corresponds to the number of terms in a document while the number of unique terms to the number of vertices in its graph-of-word representation. We give the average values per document over the entire collection.

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

Statistic	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	237	272	398	645
average # of vertices	125	157	165	185
average # of edges	608	734	901	1185

For each collection, we used a set of TREC topics (title only to mimic Web queries) and their associated relevance judgments: 51-200 for Disks 1&2 (TREC1-3 Ad Hoc Tasks), 301-450 and 601-700 for Disks 4&5 (TREC 2004 Robust

[4] http://trec.nist.gov/data/docs_eng.html

Table 2: Results for TW vs. TF scoring functions with untuned slope parameter b. Bold font marks the best performance. * (**) indicates statistical significance at p < 0.05 (0.01) using the Student's t-test with regard to the baseline (TF$_{kop}$ or BM25).

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****

Track), 451-550 for WT10G (TREC9-10 Web Tracks) and 751-850 for .GOV2 (TREC 2004-2006 Terabyte Tracks).

We evaluated the weighting models over these collections in terms of *Mean Average Precision* (MAP) and *Precision at 10* (P@10) considering only the top-ranked 1000 documents for each run. Our goal is to propose novel scoring functions that improve both metrics. The statistical significance of improvement was assessed using the Student's paired t-test rather than the Wilcoxon signed-rank test following [28, 31] recommendations for ad hoc IR. We used the R [23] implementation for the t-test (`t.test {stats}`[5]) and the output of the official `trec_eval`[6] tool as input (`-q` option to get the Average Precision for each query). We considered two-sided p-values less than 0.05 and 0.01 to reject the null hypothesis.

6.2 Platform and models

We used Terrier version 3.5 [20] to index, retrieve and evaluate the retrieval models over the TREC collections. We extended the framework to accommodate our graph-based approach (mainly graph creation, indexing of term weight instead of term frequency and novel weighting models such as TW-IDF). We chose the JUNG library [19] for graph representation and computation. We used Hadoop 1.0.3 to handle the .GOV2 dataset (426 GB). For all the datasets, the preprocessing steps involved Terrier's built-in stopword removal and Porter's stemming.

We considered four state-of-the-art scoring functions to compare with: TF-IDF, BM25, Piv+ and BM25+. They all use pivoted document length normalization with a slope parameter b set by default to 0.20 [30] for TF-IDF and its extension Piv+ and 0.75 [25] for B25 and its extension BM25+. We will present results with default value for b and tuned one using 2-fold cross-validation (up to 4 decimals, odd vs. even topic ids, MAP maximization). For TW-IDF, b does not require any tuning and a default value of 0.003 was consistently giving good results across all collections. Regarding the parameter δ defined in [14] used in the lower-bounding regularization, we did not tune it and used the default value (1.0) suggested in the original paper for both Piv+ and BM25+ [14].

6.3 Results

We present in Table 2, 4, 5 and 6 the results we obtained for each of the four tasks. We separated standard models (TF-IDF and BM25 in Table 2 and 5) and extensions (Piv+ and BM25+ in Table 4 and 6) as well as with untuned slope parameter b (Table 2 and 4) and with tuned one using cross-validation (Table 5 and 6). We indicate in Table 3 the tuned value for b that was learnt for each model on each task.

We also indicate each time the results for the TF/TW component only and then for the full model (that takes into account IDF). That way, we can see the impact of TW as an alternative to TF. Statistical significance was computed with regard to the baseline model: TF$_{kop}$ and TF$_{\delta okop}$ for TW$_p$, BM25 and BM25+ for TW-IDF (with tuning in Table 5 and 6). Note that the results for BM25 in Table 5 match the ones on the official Terrier website[7], assuring some reproducibility of the experiments.

Table 3: Values of slope parameter b tuned using cross-validation (even vs. odd topic ids).

Model	TREC1-3 Ad Hoc	TREC 04 Robust	TREC9-10 Web	TREC 04-06 Terabyte
TF$_{pol}$	0.1100	0.0635	0.0310	0.0340
TF-IDF	0.1000	0.0780	0.0340	0.0350
TF$_{kop}$	0.4180	0.3719	0.1740	0.3910
BM25	0.3600	0.3444	0.2505	0.3900
TF$_{\delta opol}$	0.1548	0.0919	0.0407	0.0430
Piv+	0.1510	0.0863	0.0340	0.0400
TF$_{\delta okop}$	0.5170	0.3780	0.2720	0.4120
BM25+	0.5510	0.3760	0.2345	0.4150

Table 2 clearly establishes the significant performance of TW-IDF over BM25 (and a fortiori TF-IDF). With tuning of the slope parameter b (Table 5), TW-IDF still outperforms BM25 on the Web datasets in terms of MAP. Moreover, tuning of parameters is known to be costly and requires a set of queries with associated relevance judgments. For a new collection of documents without such training set (like most of real-world datasets), TW-IDF appears more robust and should produce better results than BM25.

In Table 4, we compare our novel models to extensions of TF-IDF and BM25 recently proposed in [14]. Again, without tuning, TW$_p$ and TW-IDF significantly outperform the other models. This shows how well the graph-of-word encompasses concavity, document length normalization and lower-bounding regularization compared to the traditional bag-of-word. This allows our model to require less parameterization and more robustness across collections.

[5] http://stat.ethz.ch/.../t.test.html

[6] http://trec.nist.gov/trec_eval

[7] http://terrier.org/docs/v3.5/trec_examples.html

Table 4: Results for TW vs. TF+ scoring functions with untuned slope parameter b. Bold font marks the best performance. * (**) indicates statistical significance at p < 0.05 (0.01) using the Student's t-test with regard to the baseline (TF$_{\delta okop}$ or BM25+).

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$_{\delta opol}$	0.20	0.1470	0.3820	0.2002	0.3876	0.1436	0.2021	0.2055	0.5081
TF$_{\delta okop}$	0.75	0.1272	0.3240	0.2165	0.3956	0.1835	0.2354	0.2654	0.5369
TW$_p$	0.003	**0.1576****	**0.4040****	**0.2190***	**0.4133****	**0.1946****	**0.2479****	**0.2828****	**0.5407****
Piv+	0.20	0.1825	0.3813	0.2368	0.4157	0.1643	0.2438	0.2293	0.5047
BM25+	0.75	0.1558	0.3207	**0.2466**	0.4145	0.2026	0.2521	0.2830	0.5383
TW-IDF	0.003	**0.1973****	**0.4148****	0.2403	**0.4180***	**0.2125****	**0.2917****	**0.3063****	**0.5633****

In all fairness, we reported in Table 6 results for tuned Piv+ and BM25+. These are the only cases where TW-IDF performed similarly. One has to take into consideration that we are challenging a well-established model with a novel approach and in this last case, tuned state-of-the-art scoring functions. We shall see in section 7 our next ideas to further improve TW-IDF.

In Table 2, we also indicate TW that corresponds to a raw term weight without document length normalization. Surprisingly, it is already beating TF$_{pol}$, the TF component of TF-IDF. This was one of our early models and this finding encouraged us to pursue further and develop TW-IDF.

6.4 Likelihood of relevance/retrieval

We mentioned in subsection 5.3 that, *in principle*, TW-IDF should include an explicit regularization over the document length like most of the traditional scoring functions such as TF-IDF or BM25. Now we turn to seeking empirical evidence to see if this is the case in *practice*. We already witnessed in the experiments that TW$_p$ was giving much better results than the raw TW in terms of MAP and P@10.

Following Singhal et al.'s finding that a good scoring function should retrieve documents of all lengths with similar chances to their probability of relevance [29], we compared the retrieval pattern of TW-IDF (with and without regularization) against the relevance pattern extracted from the golden judgments. We followed the binning analysis strategy proposed in [29] and plotted in Figure 3 the three patterns against all document lengths on WT10G. This was developed in R with the bin size set to 5000 and the x-axis with logarithmic scale like in [14]. We also plotted the retrieval patterns of TF-IDF, Piv+, BM25 and BM25+ (with the tuned slope parameter b from Table 5 and 6).

The plot shows that TW-IDF without document length normalization ($b = 0$, blue curve, square cross points) favors more than it should longer documents and less than it should shorter documents, thus requiring pivoting and tilting ($b = 0.003$, blue curve, triangle point down points). This empirically confirms our previous analysis that TW-IDF needs document length normalization. The plot also shows clearly that TW-IDF with pivoted document length normalization matches better the likelihood of relevance (black curve, circle points) than any other retrieval functions, even BM25+ (green curve, diamond points) that was specifically designed to overcome the over-penalization for very long documents compared to BM25 (green curve, cross points) as introduced in [14]. This is yet another advantage of our scoring function TW-IDF in terms of robustness against varying document lengths.

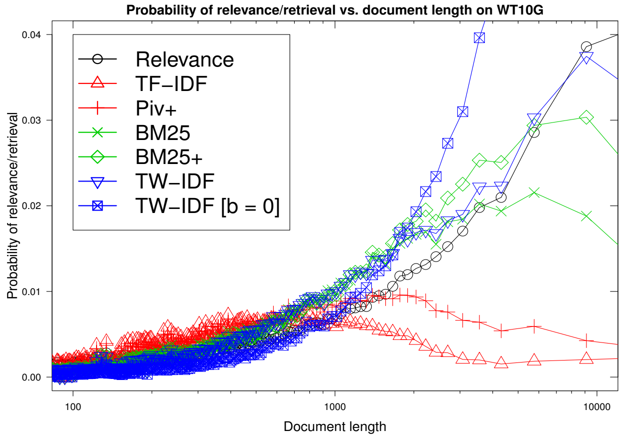

Figure 3: Comparison of retrieval and relevance patterns against all document lengths for various retrieval models on WT10G.

7. DISCUSSION AND FUTURE WORK

The proposed scoring function – TW-IDF – based on a graph-of-word representation of a document showed in practice to produce better results than TF-IDF and BM25, based on a bag-of-word representation. Moreover, it offers several advantages: no parameter tuning nor lower-bounding normalization are required due to a constant and small value for the slope parameter b (0.003) and a better robustness against varying document lengths. In the next subsections, we shall interpret this performance and discuss future work.

7.1 Number of different contexts of occurrence

Experiments showed that TW is a valid alternative to TF. TW corresponds to the indegree of a term in the graph-of-word. Since it is an unweighted graph, this means that the term weight represents the number of different contexts in which a term occurs inside a document, whatever the frequency of each context is. Rather than an overall concave term frequency, this only takes into account unique contexts of occurrence of a term and this seems to be more relevant to search. Recall that the raw term frequency was historically dampened to decrease the marginal gain of incrementing the tf value as it increases. Here, in the context of our graph representation, an additional edge is added only if the context is new. This holds the same amount of information whatever the tw value already is. This is an important find-

Table 5: Results for TW vs. TF scoring functions with tuned slope parameter b using cross-validation. Bold font marks the best performance. * (**) indicates statistical significance at p $<$ 0.05 (0.01) using the Student's t-test with regard to the baseline (TF$_{kop}$ or BM25).

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$_{pol}$	tuned	0.1555	**0.4167**	0.1946	0.3867	0.1847	**0.2729**	0.2336	0.5611
TF$_{kop}$	tuned	0.1521	0.3767	0.2176	**0.4145**	0.1915	0.2583	0.2807	**0.5785**
TW$_p$	0.003	**0.1576***	0.4040	**0.2190***	0.4133	**0.1946****	0.2479	**0.2828****	0.5407
TF-IDF	tuned	0.1936	**0.4340**	0.2261	0.4193	0.2031	0.2854	0.2589	0.5732
BM25	tuned	0.1893	0.4080	**0.2502**	**0.4382**	0.2104	**0.3210**	0.3046	**0.5899**
TW-IDF	0.003	**0.1973***	0.4148	0.2403	0.4120	**0.2125***	0.2917	**0.3063***	0.5633

Table 6: Results for TW vs. TF+ scoring functions with tuned slope parameter b using cross-validation. Bold font marks the best performance. * (**) indicates statistical significance at p $<$ 0.05 (0.01) using the Student's t-test with regard to the baseline (TF$_{\delta \circ kop}$ or BM25+).

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$_{\delta \circ pol}$	tuned	0.1481	0.3873	0.2082	0.4036	0.1949	**0.2729**	0.2463	0.5658
TF$_{\delta \circ kop}$	tuned	0.1319	0.3240	**0.2246**	**0.4185**	**0.1997**	0.2708	**0.2894**	**0.5758**
TW$_p$	0.003	**0.1576****	**0.4040****	0.2190	0.4133	0.1946	0.2479	0.2828	0.5407
Piv+	tuned	0.1841	0.3840	0.2436	0.4229	0.2117	0.2875	0.2667	0.5725
BM25+	tuned	0.1603	0.3340	**0.2547**	**0.4349**	**0.2169**	0.2771	**0.3085**	**0.5906**
TW-IDF	0.003	**0.1973****	**0.4148****	0.2403	0.4120	0.2125	**0.2917****	0.3063	0.5633

ing and to the best of our knowledge, this interpretation of the graph-of-word model has never been reported before.

7.2 Normalized weighted graph

As explained in subsection 4.3, experiments showed that the proposed framework works better at the moment without weights on the edges and this is why we reported our best performances with an unweighted graph in section 6. Intuitively, the edge weight could be the number of times the two terms co-occur. Then, the term weight would be the weighted degree. The main problem with that approach is the normalizations to apply, namely concavity and document length normalization. Similarly to the issues arising when summing the tf values in TF-IDF, the aggregation of raw edge weights leads to unbounded tw values and results in a poor overall scoring function. Because the tw value already includes this artifact when considering a simple weighted degree, it is too late to compensate it when computing the document-term score at query time. This second layer of normalizations should occur before tw is computed and thus a priori before the indexing. This is beyond the scope of the paper and should be investigated in future work. In particular, a parameterized normalization would involve the tuning at query time of a parameter set at indexing time.

7.3 TW-IDW

Through the graph-of-word representation, we challenged the *term independence assumption* made behind the bag-of-word model and proposed TW as an alternative to TF. In ad hoc IR and more generally in text mining, there is actually another assumption made: the *document independence assumption*. We commonly assume that each document in a collection is independent of one another. In particular, when computing the *document frequency* used in IDF, we consider the collection as a *bag-of-document*. In the context of a search engine, taking into account relations between documents could improve search and user experience by providing more diversity among the top results for instance. In future work, we might explore as well a *graph-of-document* representation of a collection and propose TW-IDW as an alternative to TF-IDF. Defining the *document weight* of a term in the context of a graph is a research issue in itself and is beyond the scope of this paper.

8. CONCLUSIONS

In this paper, we introduced novel document representation (graph-of-word) and retrieval model (TW-IDF) for ad hoc IR. Questioning the *term independence assumption* behind the traditional bag-of-word model, we proposed a different representation of a document that captures the relationships between the terms using an unweighted directed graph of terms. From this graph, we extract at indexing time meaningful term weights (TW) that replace traditional term frequencies (TF) and from which we define a novel scoring function, namely TW-IDF by analogy with TF-IDF. This approach led to a retrieval model that consistently and significantly outperforms BM25 and in some cases its extension BM25+ on various standard TREC datasets. In particular, experiments showed that counting the number of different contexts in which a term occurs inside a document is more effective and relevant to search than considering an overall concave term frequency in the context of Information Retrieval, a finding that has never been reported before to the best of our knowledge.

Future work might involve refining our graph-of-word model with weighted edges or proposing a scoring function (TW-IDW) that challenges the *document independence assumption* or applying this novel approach to other applications such as document summarization and phrasal indexing in the context of ad hoc IR.

9. ACKNOWLEDGMENTS

We thank the anonymous reviewers for their useful feedbacks. This material is based upon work supported by the French DIGITEO Chair grant LEVETONE.

10. REFERENCES

[1] G. Amati and C. J. Van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems*, 20(4):357–389, Oct. 2002.

[2] R. Blanco and C. Lioma. Random walk term weighting for information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, page 829–830, 2007.

[3] R. Blanco and C. Lioma. Graph-based term weighting for information retrieval. *Information Retrieval*, 15(1):54–92, Feb. 2012.

[4] F. Bonchi, R. Perego, F. Silvestri, H. Vahabi, and R. Venturini. Efficient query recommendations in the long tail via center-piece subgraphs. In *Proceedings of the 35th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '12, page 345–354, 2012.

[5] G. Erkan and D. R. Radev. LexRank: graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22(1):457–479, July 2004.

[6] H. Fang, T. Tao, and C. Zhai. A formal study of information retrieval heuristics. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '04, page 49–56, 2004.

[7] R. Ferrer i Cancho and R. Solé. The small world of human language. *Proceedings of The Royal Society of London. Series B, Biological Sciences*, 268:2261–2266, 2001.

[8] R. Ferrer i Cancho, R. Solé, and R. Köhler. Patterns in syntactic dependency networks. *Physical Review E*, 69(5):051915, 2004.

[9] G. Giannakopoulos, V. Karkaletsis, G. Vouros, and P. Stamatopoulos. Summarization system evaluation revisited: N-gram graphs. *ACM Transactions on Speech and Language Processing*, 5(3):5:1–5:39, Oct. 2008.

[10] O. Jespersen. *The Philosophy of Grammar*. Allen and Unwin, 1929.

[11] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31(1):7–15, Apr. 1989.

[12] C. Lioma and R. Blanco. Part of speech based term weighting for information retrieval. In *Proceedings of the 31st European Conference on Information Retrieval*, ECIR '09, page 412–423. Springer-Verlag, 2009.

[13] J. Liu, J. Wang, and C. Wang. A text network representation model. In *Proceedings of the 5th International Conference on Fuzzy Systems and Knowledge Discovery*, volume 4 of *FSKD '08*, page 150–154. IEEE Computer Society, 2008.

[14] Y. Lv and C. Zhai. Lower-bounding term frequency normalization. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, page 7–16, 2011.

[15] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[16] R. Mihalcea and P. Tarau. TextRank: bringing order into texts. In *Proceedings of Empirical Methods in Natural Language Processing*, EMNLP '04, 2004.

[17] R. Mihalcea, P. Tarau, and E. Figa. PageRank on semantic networks, with application to word sense disambiguation. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04. ACL, 2004.

[18] A. E. Motter, A. P. S. de Moura, Y.-C. Lai, and P. Dasgupta. Topology of the conceptual network of language. *Physical Review E*, 65(6):065102, 2002.

[19] J. O'Madadhain, D. Fisher, S. White, P. Smyth, and Y. Boey. Analysis and visualization of network data using JUNG. *Journal of Statistical Software*, 10(2):1–35, 2005.

[20] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and C. Lioma. Terrier: A high performance and scalable information retrieval platform. In *Proceedings of ACM SIGIR'06 Workshop on Open Source Information Retrieval (OSIR 2006)*, SIGIR '06, 2006.

[21] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford University, 1999.

[22] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, page 275–281, 1998.

[23] R Core Team. R: A language and environment for statistical computing, 2012. ISBN 3-900051-07-0.

[24] S. E. Robertson and K. Spärck Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3):129–146, 1976.

[25] S. E. Robertson, S. Walker, K. Spärck Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In *Proceedings of the 3rd Text REtrieval Conference*, TREC-3, page 109–126, 1994.

[26] F. Rousseau and M. Vazirgiannis. Composition of TF Normalizations: New Insights on Scoring Functions for Ad Hoc IR. In *Proceedings of the 36th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '13, 2013.

[27] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[28] M. Sanderson and J. Zobel. Information retrieval system evaluation: effort, sensitivity, and reliability. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, page 162–169, 2005.

[29] A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '96, page 21–29, 1996.

[30] A. Singhal, J. Choi, D. Hindle, D. Lewis, and F. Pereira. AT&T at TREC-7. In *Proceedings of the 7th Text REtrieval Conference*, TREC-7, page 239–252, 1999.

[31] M. D. Smucker, J. Allan, and B. Carterette. A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the 16th ACM international conference on information and knowledge management*, CIKM ’07, page 623–632, 2007.

[32] C. J. Van Rijsbergen. A theoretical basis for the use of co-occurrence data in information retrieval. *Journal of Documentation*, 33(2):106–119, 1977.

[33] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR ’01, page 334–342, 2001.