The Classic TF-IDF Vector Space Model

Abstract – This is Part 3 of an introductory tutorial series on Term Vector Theory. The classic term frequency-inverse document frequency model or TF-IDF, is discussed.

Keywords: vectors, term vector theory, term frequency, inverse document frequency, tf-idf, cosine similarity

Published: 10-27-2006; Updated: 03-19-2016

Note: This article is part of a legacy series that the author published circa 2006 at http://www.miislita.com, now a search engine site. It is now republished in pdf format here at http://www.minerazzi.com, with its content edited and updated. The original articles can be found referenced in online research publications on IR and elsewhere.

Introduction

In Parts 1 and 2 of this series on vector space models for Information Retrieval (IR) we mentioned that an index of terms is constructed by extracting n number of unique terms from a collection of D number of documents. Local, $L_{i,j}$, and global, G_i, weights can then be assigned to index terms appearing in documents and queries (Garcia, 2016a; 2016b).

For instance, the weight of an index term i present in a document j can be computed as

$$w_{i,j} = L_{i,j} G_i \tag{1}$$

A normalization weight, N_j, is some times included. Its meaning is discussed later in this tutorial.

One can find in the IR literature (Chisholm & Kolda, 1999; Lee, Chuang, & Seamons, 1997; Baeza-Yates & Ribeiro-Neto, 1999; Grossman & Frieder, 2004; Rijsbergen, 2004) different weighting schemes based on (1).

For instance, in the Binary Model (BNRY), $L_{i,j}$ is defined based on the presence or absence of an index term i in a document j, regardless of its frequency, $f_{i,j}$, and global weight.

$$w_{i,j} = L_{i,j} \begin{cases} 1 \; if \; f_{i,j} > 0 \\ 0 \; if \; f_{i,j} = 0 \end{cases} \tag{2}$$

By contrast, in the Term Count Model (FREQ) $L_{i,j}$ is defined by considering term frequencies.

$$w_{i,j} = L_{i,j} = f_{i,j} \tag{3}$$

Both models, (2) and (3), ignore global weights. Including these weights leads to a new model known as TF-IDF where global weights are computed as described below.

Global Weights

We can define the global weight of an index term across a collection of documents using probability arguments.

Let d_i be the number of documents from D that mention an index term i. Then $p_i = d_i/D$ is the probability that a document from D contains an index term i. To smoothly compare very large and small p_i values, the probability scale is compressed by taking logarithms; i.e. $log(p_i) = log(d_i/D)$. As logarithms are additive, then for any two terms $log(p_1p_2) = log(p_1) + log(p_2)$; i.e., index terms are assumed to be independent. Therefore, $p_1p_2 = d_1*d_2/D^2$.

Since $D >> d_i$, $log(d_i/D) < 0$. To avoid negative values, the ratio inside the parentheses is inverted and the result taken for a global weight, $G_i = log(D/d_i)$, now called the *inverse document frequency (IDF)*. Therefore,

$$w_{i,j} = L_{i,j}G_i = f_{i,j}log\left(\frac{D}{d_i}\right) = f_{i,j}IDF_i \tag{4}$$

where (4) is the classic *Term Frequency-Inverse Document Frequency Model* or TF-IDF Model (Salton, Wong, & Yang, 1975; Salton, 1983; Salton & Buckley, 1987).

If terms are assumed to be independent, the *IDF* assigned to, for example, a sequence of two terms should be estimated as $IDF_{12} = IDF_1 + IDF_2$. For this to be true, the presence or absence of one index term in a document or query should not be slaved to the presence or absence of another. The problem with this assumption is that often it does not hold. For example, terms relevant to a given topic tend to co-occur.

Understanding *IDF*

IDF was initially formulated by Spärck-Jones as a measure of the specificity or level of detail at which a given concept is represented by an index term (Spärck-Jones, 1972; 2004). The concept was later reformulated as a global weight in the absence of relevance information (Robertson, 2004; Spärck-Jones, Walker, & Robertson, 2000a; 2000b).

IDF provides us with a fair estimate of the discriminatory power of a term across a collection of documents. Intuitively, an index term mentioned in many documents should weigh less than one which occurs in a few documents.

For example, *a, and, in, is, of,* and *the* are low-*IDF* terms as they tend to appear in many documents. These terms are not specific to a document and cannot be used to summarize topics. They effectively cannot be used to discriminate between topics and documents. Conversely, brand names, technical terms, and scientific nomenclature words tend to be more discriminatory, then high-*IDF* terms.

TF-IDF Based Models

Nowadays, a family of weighting schemes based on (4) can be found in the IR literature (Salton & Buckley, 1987; Lee, Chuang, & Seamons, 1997; Chisholm & Kolda, 1999) by modifying $L_{i,j}$ and G_i . For instance, some times $L_{i,j}$ is written by normalizing and then augmenting term frequencies, like this:

$$L_{i,j}\begin{cases}\frac{f_{i,j}}{max f_{i,j}}\ if\ f_{i,j} > 0\\ 0\ if\ f_{i,j} = 0\end{cases} \tag{5}$$

$$L_{i,j}\begin{cases}0.5 + 0.5\frac{f_{i,j}}{max f_{i,j}}\ if\ f_{i,j} > 0\\ 0\ if\ f_{i,j} = 0\end{cases} \tag{6}$$

where $maxf_{i,j}$ is the maximum frequency of any index term in document j. Expression (6) is known as the augmented normalized frequency model or ATF1 where for $f_{i,j} > 0$ the model dampens down local weights to $0.5 > L_{i,j} \geq 1$.

Replacing $L_{i,j}$ in (4) with (5) and (6) leads, respectively, to two new TF-IDF schemes:

$$w_{i,j} = \frac{f_{i,j}}{max\, f_{i,j}} log\left(\frac{D}{d_i}\right) = \frac{f_{i,j}}{max\, f_{i,j}} IDF_i \quad (7)$$

and

$$w_{i,j} = \left(0.5 + 0.5\frac{f_{i,j}}{max\, f_{i,j}}\right) log\left(\frac{D}{d_j}\right) = \left(0.5 + 0.5\frac{f_{i,j}}{max\, f_{i,j}}\right) IDF_i \quad (8)$$

As a query is like another document, similar expressions can be formulated for computing $w_{i,q}$. In addition, document and query weighting schemes can be combined. Actually, Salton tried about 1800 combinations of which 287 were found to be distinct (Salton & Buckley, 1987).

Which combination should then be used? The answer depends on many factors, like users search behaviors, document and query lengths, and the nature of the database. For instance, Web searchers are not like users working in an IR computer lab under controlled conditions.

Average Web searchers tend to use short queries consisting of few terms, also being short and frequently consisting of nouns. They are not prone to use thesauri or lookup lists, nor they are inclined to search using rare terms they might never heard of. Instead, they are prone to reformulate queries with previously used terms or derivative of these.

Average Web users also query unstructured databases like those belonging to commercial search engines. Such database collections are not static, but always changing. Many of these are plagued with problems not found in an IR computer laboratory.

And there is still the problem of documents already indexed and that were designed for the sole purpose of spamming a search engine index, a practice known as *spamdexing* (AIRWeb, 2007). Thus, when it comes to Web searches, there is no such thing as an "optimal" combination of document and query weighting schemes. Regardless of the combination of weighting schemes used for documents and queries, once the $w_{i,j}$ and $w_{i,q}$ weights are calculated, document and query vectors are computed and compared using either their similarity coefficient, *coeff*(d_j, *q*), defined as the dot product between the vectors,

$$coeff(d_j, q) = \mathbf{d_j} \bullet \mathbf{q} = \sum_{i=1}^{n} w_{i,j}\, w_{i,q} \tag{9}$$

or in terms of their *cosine similarity*, *sim(d_j, q)*; i.e., the cosine of the angle between vectors,

$$sim(d_j, q) = \frac{\mathbf{d_j} \bullet \mathbf{q}}{\|\mathbf{d_j}\|\,\|\mathbf{q}\|} = \frac{\sum_{i=1}^{n} w_{i,j}\, w_{i,q}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2}\sqrt{\sum_{i=1}^{n} w_{i,q}^2}} \tag{10}$$

where $\mathbf{d_j}$ and $\mathbf{q}$ are vectors with absolute magnitudes $\|\mathbf{d_j}\|$ and $\|\mathbf{q}\|$. A geometric analysis of these two similarity measures and few others is available (Jones and Furnas, 1987). In the early IR literature, a distinction between (9) and (10) was made by defining the normalization factor

$$N_j = \frac{1}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2}\sqrt{\sum_{i=1}^{n} w_{i,q}^2}} \tag{11}$$

and then using

$$w_{i,j} = L_{i,j} G_i N_j \tag{12}$$

where (9) is obtained by setting $N_j = 1$ and (10) with $N_j = \frac{1}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2}\sqrt{\sum_{i=1}^{n} w_{i,q}^2}}$.

The latter is the so-called cosine normalization which can also be obtained by converting vectors to unit vectors before computing dot products. Nowadays (10), *sim(d_j, q)*, is mostly used regardless of how the weights are computed.

In the next section we present a working example where documents are ranked using both similarity measures. Document and query terms are weighted with (4), the classic TF-IDF Model. The example is taken from page 15 of the book *Information Retrieval: Algorithms and Heuristics* (Grossman & Frieder, 2004) where these authors used (9), $coeff(d_j, q)$.

TF-IDF Calculations Example

A collection consisting of three "documents" ($D = 3$) is searched for the query [gold silver truck].

d_1 = Shipment of gold damaged in a fire.

d_2 = Delivery of silver arrived in a silver truck.

d_3 = Shipment of gold arrived in a truck.

To construct the index of terms, the documents were processed as follows:

- Tokenization: Punctuation removed and text lowercased.
- Filtering: None. Stopwords were not removed.
- Stemming: None. Terms were not reduced to their roots.

Table 1 lists index terms and their raw data.

Table 1. Index terms with raw data.

Index terms	q		d_1	d_2	d_3		d_i	IDF_i
a	0		1	1	1		3	0.00
arrived	0		0	1	1		2	0.18
damaged	0		1	0	0		1	0.48
delivery	0		0	1	0		1	0.48
fire	0		1	0	0		1	0.48
gold	1		1	0	1		2	0.18
in	0		1	1	1		3	0.00
of	0		1	1	1		3	0.00
silver	1		0	2	0		1	0.48
shipment	0		1	0	1		2	0.18
truck	1		0	1	1		2	0.18

Table 2 shows the result of computing term weights with (4), TF-IDF.

Table 2. Index term weights.

Index terms	$w_{i,q}$		$w_{i,1}$	$w_{i,2}$	$w_{i,3}$
a	0.00		0.00	0.00	0.00
arrived	0.00		0.00	0.18	0.18
damaged	0.00		0.48	0.00	0.00
delivery	0.00		0.00	0.48	0.00
fire	0.00		0.48	0.00	0.00
gold	0.18		0.18	0.00	0.18
in	0.00		0.00	0.00	0.00
of	0.00		0.00	0.00	0.00
silver	0.48		0.00	0.95	0.00
shipment	0.00		0.18	0.00	0.18
truck	0.18		0.00	0.18	0.18

Table 3 shows that *coeff(d_j, q)* results agree with those of Grossman and Frieder.

Table 3. Documents-query similarity results.

Measure	Datum	Formula	d_1	d_2	d_3
Similarity Coefficient	$\mathbf{d_j} \bullet \mathbf{q}$	$coeff(d_j, q) = \sum_{i=1}^{n} w_{i,j}\, w_{i,q}$	0.031	0.486	0.062
$\sum_{i=1}^{n} w_{i,q}^2$	0.29	$\sum_{i=1}^{n} w_{i,j}^2$	0.52	1.20	0.12
$\sqrt{\sum_{i=1}^{n} w_{i,q}^2}$	0.54	$\sqrt{\sum_{i=1}^{n} w_{i,j}^2}$	0.72	1.10	0.35
Magnitude Product	$\|\mathbf{d_j}\|\, \|\mathbf{q}\|$	$\sqrt{\sum_{i=1}^{n} w_{i,j}^2}\, \sqrt{\sum_{i=1}^{n} w_{i,q}^2}$	0.39	0.59	0.19
Cosine Similarity	$\frac{\mathbf{d_j} \bullet \mathbf{q}}{\|\mathbf{d_j}\|\, \|\mathbf{q}\|}$	$sim(d_j, q) = \frac{\sum_{i=1}^{n} w_{i,j}\, w_{i,q}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2}\, \sqrt{\sum_{i=1}^{n} w_{i,q}^2}}$	0.08	0.82	0.33

Using either *coeff(d_j, q)* or *sim(d_j, q)*,the documents rank in the same order; i.e., $d_2 > d_3 > d_1$.

A Linear Algebra Approach

In previous tutorials, we presented a linear algebra approach that greatly simplifies vector space calculations (Garcia, 2016b; 2016c).

Essentially, from Table 2, document and query vectors are converted to unit vectors, denoted with a hat (^), and the **q**, **A**, and $\mathbf{q}^T\mathbf{A}$ matrices computed.

Index terms	$\hat{q}$	$\hat{d_1}$	$\hat{d_2}$	$\hat{d_3}$
a	0.00	0.00	0.00	0.00
arrived	0.00	0.00	0.16	0.50
damaged	0.00	0.66	0.00	0.00
delivery	0.00	0.00	0.44	0.00
fire	0.00	0.66	0.00	0.00
gold	0.33	0.24	0.00	0.50
in	0.00	0.00	0.00	0.00
of	0.00	0.00	0.00	0.00
silver	0.89	0.00	0.87	0.00
shipment	0.00	0.24	0.00	0.50
truck	0.33	0.00	0.16	0.50

q = | **A** =

$\mathbf{q}^T\mathbf{A}$ =

d_1	d_2	d_3
0.08	0.82	0.33

Because unit vectors are used, $\mathbf{q}^T\mathbf{A}$ is a matrix of cosine similarities equal to dot products; i.e., $sim(d_j, q) = coeff(d_j, q)$ so any geometric-based differences relevant to the retrieval problem (Jones & Furnas, 1987) disappear. As expected the ranking order is not affected.

Finally, if we want to compute document-query and document-document cosine similarities in one step, we can use the matrix augmentation technique introduced in Part 2 of this series (Garcia, 2016b). The technique consists in augmenting **A** with the unit vector of the query and computing the $\mathbf{A}^T\mathbf{A}$ matrix. As noted before, $\mathbf{A}^T\mathbf{A}$ is a matrix that stores cosine similarities equal to dot products, $sim(d_j, q) = coeff(d_j, q)$; i.e.

Index terms		$\hat{q}$	$\hat{d_1}$	$\hat{d_2}$	$\hat{d_3}$
a		0.00	0.00	0.00	0.00
arrived		0.00	0.00	0.16	0.50
damaged		0.00	0.66	0.00	0.00
delivery		0.00	0.00	0.44	0.00
fire		0.00	0.66	0.00	0.00
gold	**A** =	0.33	0.24	0.00	0.50
in		0.00	0.00	0.00	0.00
of		0.00	0.00	0.00	0.00
silver		0.89	0.00	0.87	0.00
shipment		0.00	0.24	0.00	0.50
truck		0.33	0.00	0.16	0.50

	q	d_1	d_2	d_3	
	1.00	0.08	0.82	0.33	q
$\mathbf{A^TA}$ =	0.08	1.00	0.00	0.24	d_1
	0.82	0.00	1.00	0.16	d_2
	0.33	0.24	0.16	1.00	d_3

Notice that the first row and column cells of $\mathbf{A^TA}$ store ranking results while non-diagonal cells document-document cosine similarities. A straightforward comparison between documents is now possible. In this example,

$sim(d_1, d_2) = 0.00$
$sim(d_1, d_3) = 0.24$
$sim(d_2, d_3) = 0.16$

Clearly the fact that d_2 and d_3 are more similar to q than d_1 does not necessarily mean that d_3 is more similar to d_2 than to d_1. In this example, d_1 and d_3 are the most similar documents.

Limitations of the TF-IDF Model

The TF-IDF model suffers of severe limitations; i.e.

- It can be gamed via $L_{i,j}$ weights, i.e., by simply repeating terms (*keyword stuffing*).
- *IDF* weights are not based on relevance information, but on merely matching terms.
- Documents sharing high-order co-occurrence and that might be relevant are ignored.
- A matrix must be recomputed each time a new document is added to a collection.

Perhaps the most severe limitation is the term independence assumption made with TF-IDF based models. According to this assumption, documents are *bags of words* where terms occur by chance and where their order does not matter.

Frequently that is not the case. Terms can be dependent due to

- **Polysemy**; i.e., same terms can be used in different contexts
  Example*:* [driving cars] vs. [driving results]
  Thus, irrelevant documents can be retrieved because they may share some words from the query. This affects precision.
- **Synonymity**; i.e., different terms can be used in the same contexts
  Example: [car insurance] vs. [auto insurance]
  So relevant documents might not be retrieved. This affects recall.
- **Ordering**; i.e, different terms can be used in different positions in different contexts
  Example: [junior college] vs. [college junior]
  This can affect precision and recall.

Conclusion

The classic term frequency-inverse document frequency model or TF-IDF has been discussed. The model is based on local and global weights, with the latter being defined using the notion of the specificity of terms.

As *IDF* is a weighting scheme in the absence of relevance information and based on term matching, it suffers of severe limitations common to many vector space models. In upcoming tutorials, we discuss several modifications and workarounds that have been incorporated to the family of TF-IDF models.

Exercises

1. Rework this tutorial exercise, this time using the following TF-IDF models to score both document and query terms. Compare results. See (5) and (6).

   - $w_{i,j} = \left(\frac{f_{i,j}}{max f_{i,j}}\right) IDF_i$
   - $w_{i,j} = \left(0.5 + 0.5\frac{f_{i,j}}{max f_{i,j}}\right) IDF_i$

2. Rework this tutorial exercise, this time by defining $L_{i,j}$ in (4) with the local weight models known as LOGA and LOGN, where $avef_{i,j}$ is the average term frequency in document j. Compare results. See Chisholm & Kolda (1999).

   - LOGA: $L_{i,j} \begin{cases} 1 + \log(f_{i,j}) \; if \; f_{i,j} > 0 \\ 0 \; if \; f_{i,j} = 0 \end{cases}$
   - LOGN: $L_{i,j} \begin{cases} \frac{1 + \log(f_{i,j})}{1 + \log(avef_{i,j})} \; if \; f_{i,j} > 0 \\ 0 \; if \; f_{i,j} = 0 \end{cases}$

References

AIRWeb (2007). Adversarial Information Retrieval on the Web. Retrieved from http://airweb.cse.lehigh.edu/2007/cfp.html

Baeza-Yates, R. and Ribeiro-Neto, B. (1999). Modern Information Retrieval. Adisson Wesley. Book Review. Retrieved from http://www.amazon.com/gp/customer-reviews/R2HC8ULDSMXKZQ/ref=cm_cr_arp_d_rvw_ttl?ie=UTF8&ASIN=020139829X

Chisholm, E. and Kolda, T. G. (1999). New Term Weighting Formulas for the Vector Space Method in Information Retrieval. Oak Ridge National Laboratory. Retrieved from http://www.sandia.gov/~tgkolda/pubs/pubfiles/ornl-tm-13756.pdf

Jones, W. P. and Furnas, G. W. (1987). Pictures of Relevance: A Geometric Analysis of Similarity Measures. JASIS, 38(6), 420-442. Retrieved from http://furnas.people.si.umich.edu/Papers/PicturesOfRelevance.pdf

Garcia, E. (2016a). Term Vector Theory and Keyword Weights. Retrieved from http://www.minerazzi.com/tutorials/term-vector-1.pdf

Garcia, E. (2016b). The Binary and Term Count Models. Retrieved from http://www.minerazzi.com/tutorials/term-vector-2.pdf

Garcia, E. (2016c). A Linear Algebra Approach to the Vector Space Model. Retrieved from http://www.minerazzi.com/tutorials/term-vector-linear-algebra.pdf

Grossman, D. A. and Frieder, O. (2004). Information Retrieval: Algorithms and Heuristics. Springer.

Book Review. Retrieved from http://www.amazon.com/review/RACNGPXD2GNE7/ref=cm_cr_dp_title?ie=UTF8&ASIN=1402030045&channel=detail-glance&nodeID=283155&store=books

Lee, D. L., Chuang, H., and Seamons (1997). Document Ranking and the Vector-Space Model. IEEE March/April, pp 67-75. Retrieved from http://www.cs.ust.hk/faculty/dlee/Papers/ir/ieee-sw-rank.pdf

Rijsbergen, K. (2004). The Geometry of Information Retrieval. Cambridge University Press, UK.

Book Review. Retrieved from http://www.amazon.com/review/R3FM04FS4ZDHGC/ref=cm_cr_dp_title?ie=UTF8&ASIN=0521838053&channel=detail-glance&nodeID=283155&store=books

Robertson, S. E. (1972). Term Specificity. Reprinted from Letter and reply, Journal of Documentation 28, 164-165. Retrieved from http://www.staff.city.ac.uk/~sb317/idfpapers/letters.pdf

Robertson, S.E. (1974). Specificity and weighted retrieval. Journal of Documentation 30, 41-6.

Robertson, S. E. (2004). Understanding Inverse Document Frequency: On Theoretical Arguments for IDF. Reprinted from Journal of Documentation 60, 503-520. Retrieved from http://www.staff.city.ac.uk/~sb317/idfpapers/Robertson_idf_JDoc.pdf

Robertson, S.E. and Spärk-Jones, K. (1994). Simple, proven approaches to text retrieval. Reprinted from University of Cambridge Computer Laboratory Technical Report no. 356, 1994 (updated 1996,1997,2006). Retrieved from http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-356.pdf

Robertson, S.E. and Spärk-Jones, K. (1976). Relevance weighting of search terms. Journal of the American Society for Information Science 27, 129-46 (1976). Reprinted in: P. Willett (ed.), Document Retrieval Systems. Taylor Graham, 1988. (pp 143-160). Retrieved from http://www.staff.city.ac.uk/~sb317/papers/RSJ76.pdf

Salton, G. (1983). *Introduction to Modern Information Retrieval*. McGraw-Hill.

Salton, G. and Buckley, C. (1987). Term Weighting Approaches in Automatic Text Retrieval.87-881. Cornell University. Retrieved from https://ecommons.cornell.edu/bitstream/handle/1813/6721/87-881.pdf?sequence=1&isAllowed=y

Salton, G., Wong, A., and Yang, C. S. (1975). A Vector Space Model for Automatic Indexing. Communications of the ACM 18 (11): 613. Retrieved from http://elib.ict.nsc.ru/jspui/bitstream/ICT/1230/1/soltan_10.1.1.107.7453.pdf
see also http://www.bibsonomy.org/bibtex/10a4c67f15a4869634d8e5e39ba3e7113

Salton, G. and Yang, C. S. (1973). On the Specification of Term Values in Automatic Indexing. TR 73-173, Cornell University. Retrieved from https://ecommons.cornell.edu/bitstream/handle/1813/6016/73-173.pdf?sequence=1&isAllowed=y

Spärk-Jones, K. (1972). A Statistical Interpretation of Term Specificity and its Application in Retrieval. Journal of Documentation, Vol 60, 5, 493-502. Retrieved from http://www.staff.city.ac.uk/~sb317/idfpapers/ksj_orig.pdf

Spärck-Jones, K. (2004). IDF term weighting and IR research lessons. Reprinted from Journal of Documentation 60, 5, 521-523. Retrieved from http://www.staff.city.ac.uk/~sb317/idfpapers/ksj_reply.pdf

Spärck-Jones, K., Walker, S., and Robertson, S.E. (2000a). A probabilistic model of information retrieval: development and comparative experiments. Information Processing and Management 36, Part 1 779-808. Retrieved from http://www.staff.city.ac.uk/~sb317/blockbuster/pmir-pt1-reprint.pdf

Spärck-Jones, K., Walker, S., and Robertson, S.E. (2000b). A probabilistic model of information retrieval: development and comparative experiments. *Information Processing and Management* **36**, Part 2 809-840. Retrieved from http://www.staff.city.ac.uk/~sb317/blockbuster/pmir-pt2-reprint.pdf