

Linear Time Series Models for Term Weighting in Information Retrieval

Miles Efron
Graduate School of Library and Information Science
University of Illinois
501 E. Daniel St.
Champaign, IL 61820
`mefron@illinois.edu`

September 25, 2009

Abstract

Common measures of term importance in information retrieval (IR) rely on counts of term frequency; rare terms receive higher weight in document ranking than common terms receive. However, realistic scenarios yield additional information about terms in a collection. Of interest in this paper is the temporal behavior of terms as a collection changes over time. We propose capturing each term’s collection frequency at discrete time intervals over the lifespan of a corpus and analyzing the resulting time series. We hypothesize the collection frequency of a term x at time t is predictable by a linear model of the term’s prior observations. On the other hand, a linear time series model for a strong discriminators’ collection frequency will yield a poor fit to the data. Operationalizing this hypothesis, we induce three time-based measures of term importance and test these against state-of-the-art term weighting models.

1 Introduction

When we process a corpus of text for information retrieval (IR) a fundamental activity is term weighting. The term weighting problem involves gauging the usefulness of each index term as a topical discriminator. Following the intuition that in a query such as *the great Recession* the appearance in a document of the word *the* is less predictive of relevance than is an instance of *recession*, term weighting models typically assign numeric scores to each word in the indexing vocabulary, where these scores serve the purpose of lending “important” words greater impact on document ranking than less important words exert.

This paper proposes a novel approach to term weighting. We offer a method of analyzing documents in corpora to derive a numerical weight for each word to be used during document ranking. The novelty in our approach lies in the motivation and the method used for calculating these term weights.

Our work begins with the fact that in most IR situations corpora change over time. That is, as time passes documents are added (or deleted) from the collection. As the IR system’s index changes over time, the frequency of various index terms also changes. Thus the frequency of *recession* at time t might be, say, 1000, while at time $t + 1$ its frequency might be 1010. These changes are informative.

The premise of this paper is that the way a term’s collection frequency changes over time lends useful information about that term’s power as a discriminator. Specifically, we hypothesize that the behavior of weak discriminators is easily described by a simple linear time series model, while useful discriminators’ distribution over time is too erratic to describe faithfully with a linear model. We argue that the collection frequency of a term x at time t is effectively modeled by a linear function of the p previously observed values $x_{t-1}, x_{t-2}, \dots, x_{t-p}$. On the other hand, we argue that the collection frequency of strong discriminators will see a poor fit to a linear model $x_t = \mu + \beta_0 + \beta_1 x_{t-1} + \beta_2 x_{t-2} + \dots + \beta_p x_{t-p}$.

2 Context of this Research

This paper’s central argument is that a term’s collection frequency over time bears on its value as a discriminator. This argument situates our work in the context of a great deal of prior work on temporal approaches to IR-related tasks. In this section we detail how our work relates to the landscape of temporally-informed IR research.

Among the earliest problems that use temporal data in IR is topic detection and tracking (TDT) (Allan, 2002; Swan and Allan, 1999). The TDT community has developed a rich literature and robust models for identifying and providing information on topics in temporal data such as news feeds. TDT approaches are inherently temporal, often creating timelines (Swan and Allan, 2000) and other time-based overviews by recourse to statistical time series analysis (Lavrenko et al., 2000; Guralnik and Srivastava, 1999).

While topic detection and tracking has evolved into a field in its own right, temporal approaches to organizing text have branched out in other directions, too. Recent work has formulated a more general time-based approach to textual data. Miller et al. (1998) refer to this work as temporal text mining (TTM). TTM entails activities such as summarization, trend analysis, and description of topic evolution. A good deal of recent attention treats the “burstiness” of topics over time (Kleinberg, 2002; Wang et al., 2007; Wang and McCallum, 2006). That is, scholars working in this area model topics temporally with an eye towards finding interesting, useful, or otherwise important ideas.

An ongoing body of TTM literature concerns tracking “memes” (transitory topics expressed linguistically) over dynamic media such as the Web, or more specifically, blogs, forums, etc. (Leskovec et al., 2009) give an excellent overview of this task, as well as a detailed approach to meme-tracking. This line of work brings temporal analysis to bear on problems such as recommendation (Nguyen et al., 2008), but also for more general mapping of an information space (Kumar et al., 2004; Cokol and Rodriguez-Esteban, 2008).

Gruhl et al. (2004) describe the way in which topics evolve on the Internet. Gruhl et al. identify two types of topics—*chatter* and *spike* topics. Chatter topics persist over time, while spike topics appear in bursts over short periods. This distinction is similar to our distinction between strong and weak discriminators. As in (Gruhl et al., 2004) we distinguish between terms whose collection frequency follows a consistent, simple pattern, and those whose appearance is erratic.

With respect to information retrieval per se, time has informed a variety of studies. For instance Li and Croft (2003) incorporate temporality into the language modeling framework. This work focuses on improving performance for queries that are themselves time-bound, where most relevant documents were published in a particular temporal window. Li et al.’s method uses the temporal nature of these queries to induce time-based language models for document ranking. Using a different approach, Metzler et al. tackle a similar problem in (Metzler et al., 2009).

Time has also been used to classify queries, with an eye towards improving their results or to predicting the quality of their results. Rose and Diaz have proposed analyzing the temporal distribution of documents

returned for queries, arguing that different types of queries fall into distinct classes in this context. In (Jones and Diaz, 2007) and (Diaz and Jones, 2004) this approach yields information about queries that is useful for IR.

To our knowledge the idea closest to this paper’s work is (Liebscher and Belew, 2003). Liebscher and Belew (2003) describe an approach that they call “temporal term weighting.” However, their work is conceptually distinct from ours. Liebescher and Belew argue that terms whose frequency is high early in the development of a corpus should bear more weight than late-comer terms. This is in contrast to the premise of this paper that a term’s distribution over the lifetime of the corpus indicates its discriminative value. Additionally, Liebescher’s and Belew’s temporal term weighting is offered as a possible approach to IR; the authors do not implement or test the method.

In some sense the present paper is less sophisticated than the work cited in this section. Whereas the authors cited here are concerned with the temporal distribution of *topics* we analyze only simple unigram *terms*. More substantively, we are not concerned with precisely *when* a given term occurs over the analyzed time span. For our purposes, a term’s time-based weight is a function of its behavior over the entire history of the index.

Most basically, our work differs from previous research insofar as we capitalize on temporal term behavior specifically for the problem of term weighting in IR. News recommendation, meme tracking, topic detection, etc. all form problems cognate to IR. But to the best of our knowledge this work is novel—while strongly informed by prior work—in its application of time to the term weighting problem.

Aside from temporal concerns, our work is also situated in the literature of term weighting for IR. This is of course among the oldest and most well-studied branches of IR. Many traditional IR models rank documents against a query using a score that is derived using an equation following the structure of Eq. 1. Given a query Q and a document D we have the score:

$$S(D, Q) = \sum_{x \in Q} w_{xQ} \cdot w_{xD} \cdot w_{xC} \quad (1)$$

where S is the document’s “score” against the query, w_{xQ} is the weight of term x in the query, w_{xD} is the weight of term x in the document, and w_{tC} is a global weight (i.e. the collection weight) for the term.

This paper is concerned with the global term weight, w_{xQ} in Eq. 1. That is, in the following section we will develop a novel approach to deriving these weights. Because these weights are query-independent they can be computed at index time.

However, assigning high weight to “important” terms and low weight to unimportant terms is a fundamental problem in IR. This sort of weighting has a literature of its own Salton and Buckley (1987); Dubin (1999). The mainstay of collection-level term weights is the familiar inverse document frequency (IDF):

$$IDF(x) = \log \frac{N}{n(x)} \quad (2)$$

where N is the number of documents in the corpus and $n(x)$ is the number of documents containing term x (i.e. its document frequency).

IDF has been studied widely (Jones, 1979), finding a variety of mathematical interpretations (cf. (Wu et al., 2008; Roelleke and Wang, 2008; Roelleke, 2003; Aizawa, 2003; Hiemstra, 2000)). And while IDF is especially well known, similar term importance measures inform most state-of-the-art IR models. Methods such as the language modeling approach induce an IDF-like weight during the model smoothing process (Lafferty and Zhai, 2001). Likewise, in the absence of relevance information, IDF informs document ranking under the probabilistic BM25 algorithm (Robertson et al., 1995).

3 Time Series Data in IR

A full treatment of time series analysis is beyond the scope of this work, and readers are referred to the excellent treatments of the topic provided in (Montgomery et al., 2008) and (Shumway and Stoffer, 2006) *inter alia*. Our own notation, and much of the exposition in this section follows Shumway and Stoffer (2006).

Let $X = \{x_1, x_2, \dots, x_n\}$ be a time series of length n . Each of the n data points that comprise X corresponds to an observation of a variable at a discrete moment in time. That is x_t is (for our purposes) a real-valued number, for the integer $t \in 1 \dots n$.

Time series data is common in a variety of settings such as finance (where we might measure the value of an asset at the close of each trading day), ecology (where the surface temperature of the ocean could be measured at regular intervals), and signal processing (where tonal variation is captured as frequencies measured at very close intervals).

In information retrieval, we may admit similar analysis by considering the way a corpus changes over time. That is, at time t the corpus contains, say, N_t documents, C_t word tokens, and term x occurs x_t times. However, at another time t' each of these counts might be different. Often, as time passes, documents are added to (or deleted from) a collection; the overall word counts change with this growth, and the collection frequencies of individual words also change. We argue that these changes are useful for IR.

As an example, consider Figure 1. The data in Figure 1 come from the LA Times corpus, collected by NIST in support of the Text REtrieval Conference (TREC)¹. The LA Times data consists of 131,896 news articles—those article published in the *Los Angeles Times* during 1989 and 1990.

To construct Figure 1 we examined the collection at weekly intervals, which form the figure’s x -axis. That is, the leftmost portion of each panel represents week 1, while the rightmost portion represents week 104. The two panels that comprise the figure are word counts captured at these weekly intervals. The left panel shows the number of times the word *the* occurs, and the right panel gives the frequency of the term *schengen* (*schengen* was chosen simply for purposes of exposition).

Common term weighting methods such as IDF are concerned with the magnitude of each term’s frequency, that is, its value on the y -axis of Figure 1. Clearly, *the* occurs much more often (at week 104) than does *schengen*, suggesting that *schengen* should be given greater weight in document ranking.

But in addition to the magnitude of a term’s collection frequency, Figure 1 suggests that *the* and *schengen* change over time in very different ways. The graph for *the* is strongly linear. On the other hand, *schengen* grows in fits and starts.

While it goes without saying that the magnitude of a term’s collection frequency is informative with respect to its discriminative power, our argument is that the shape of a term’s growth pattern over time is also informative in this regard. Thus the remainder of this paper concerns analyzing the time series X for a given term x . In this context x_t gives the collection frequency of term x at time t in the evolution of the collection.

3.1 Transformations Applied to Time Series

Before deriving term weights from time series, a few transformations to the raw term count series will be useful.

¹<http://trec.nist.gov>

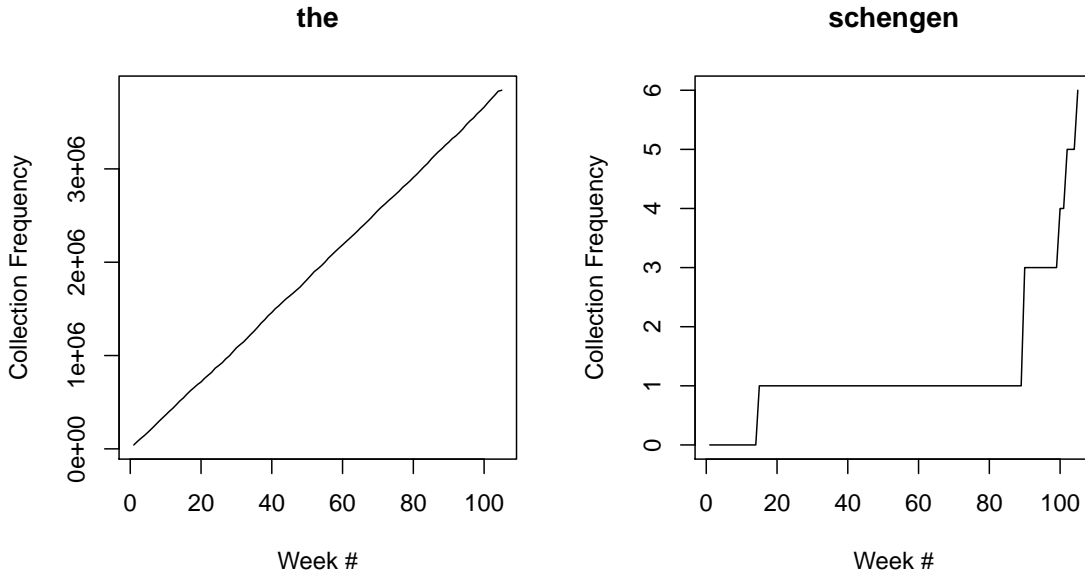


Figure 1: Time series for two words from the LA Times TREC data. The x -axis is the week at which each observation was made. The y -axis gives the collection frequency of each term at the corresponding moment in time.

First, in order to assure that we are analyzing the temporal behavior of each term, as opposed to its simple magnitude of collection frequency, we normalize each term’s time series. If χ is the time series containing the raw collection frequencies for term x , we will use the transformation:

$$X = \frac{1}{\sqrt{\sum_{i=1}^n \chi^2}} \chi. \quad (3)$$

If we consider the time series χ as an n -vector, this simply normalizes each term’s time series to unit length. Unless otherwise stated, in the discussion that follows, we assume that all time series have been normalized in this way.

A second transformation aims to “de-trend” the data. The series in Figure 1 show clear upward trends as time passes. The autoregressive models described below assume that the data lacks such a trend². These models assume that the variance and the expected value of X is the same for all times $t \in 1 \dots n$. This is an informal definition of stationarity.

A simple but effective means of transforming trended data to a form that is closer to stationarity is by “differencing” the series. A differenced time series has the form

$$x'_t = x_t - x_{t-\ell} \quad (4)$$

where ℓ is an integer, the so-called lag of the differencing operation. The differenced series X' is of length $n - \ell$ where n is the length of the original series. Throughout this paper, when we describe a differenced series we use $\ell = 1$. We find that changing ℓ has very little effect on the data we analyze because $\ell = 1$ de-trends our data very effectively, leaving little influence for higher-order lags.

²More formally, the autoregressive model assumes that the time series X is *stationary*. Stationarity is not crucial to our discussion, and in the interest of brevity we omit a treatment of it. Readers may consult (Shumway and Stoffer, 2006, Ch. 1) *inter alia* for a full treatment of stationarity.

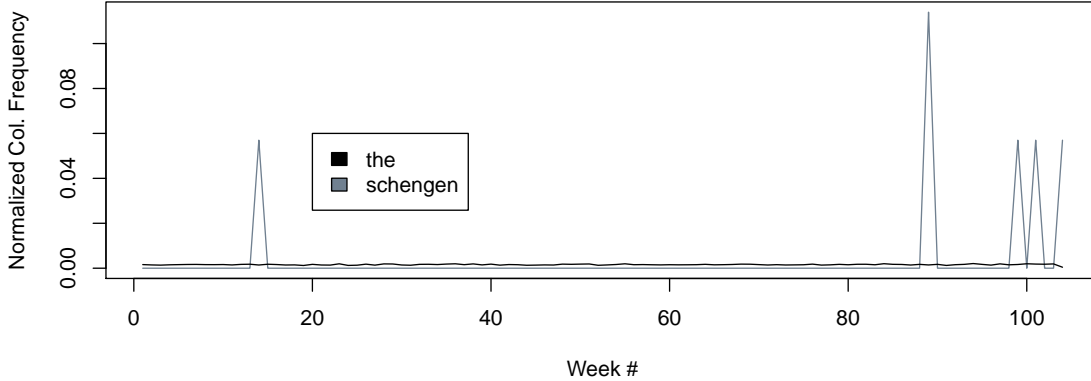


Figure 2: Differenced ($\ell = 1$) time series for terms *the* and *schengen* on the LA Times data, split into weekly intervals. The y -axis is the collection frequency of each term, normalized using Eq. ??

Figure 2 shows the data given from Figure 1 after differencing. It is clear from the figure that the upward trend in the raw data has been removed by the differencing operation. While it is not guaranteed that differencing gives a stationary time series, differencing suffices for the purposes of this paper.

4 Linear Models for Time Series

Intuitively, Figure 2 suggests that powerful discriminators’ time series will look different from the series for weakly discriminative terms. In this section we will make this intuition more precise. This section concerns models of a time series X that take the form:

$$x_t = \mu + \beta_0 + \beta_1 x_{t-1} + \beta_2 x_{t-2} + \dots + \beta_p x_{t-p} + \epsilon_t \quad (5)$$

where μ and $\beta_0, \beta_1, \beta_2, \dots, \beta_p$ are the model parameters, p is the “order” of the model, and ϵ is a finite-variance iid random variable. We shall refer to the model in Eq. 5 as a *weakly autoregressive model*. The model is autoregressive insofar as it models x_t as a function of previous observations of X . That is, we use the data in X to predict itself. We term Eq. 5 *weakly* autoregressive because it is more general than the formal autoregressive models described below. In particular, we do not assume stationarity.

Intuitively, the autoregressive model of Eq. 5 states that the value of X at time t is a linear function of the p previously observed observations on X , plus a random error term.

We argue that Eq. 5 models highly discriminative terms poorly; their behavior is not linear on the p prior observations. On the other hand, a weak discriminator’s collection frequency at time t follows Eq. 5 closely. Thus our central argument is that a term’s weight should be inversely proportional to the goodness of fit of a weakly autoregressive model.

The motivation behind this argument is that the collection frequency of weak discriminators will change in a consistent way for an arbitrary pair of observations t and $t + 1$. A weak discriminator entails roughly a constant proportion of the vocabulary across time. On the other hand, more valuable terms’ collection

frequency is prone to dramatic jumps; its values at time $t-1, t-2, \dots, t-p$ are not predictive of its collection frequency at time t .

The following subsections describe three simple linear models for time series data. Each model is concerned with predicting (or estimating) x_t in a series X by analysis of the series' previous observations. That is, if X is linear on its prior observations, we should be able to induce a model of x_t as a linear function of $x_{t-1}, x_{t-2}, \dots, x_{t-p}$ for some integer p .

4.1 Moving Averages

The simplest model we treat here is the moving average. Moving averages are widely used in a variety of fields, especially in econometrics where they serve as smoothing functions of financial data as described by Tsay (2005). The moving average is a convolution filter of the series X . It is not a proper probabilistic model with a generative underpinning such as the autoregressive model described in the following subsection. Thus it is non-parametric, but also lacks a principled measure of goodness of fit.

A moving average of order p approximates each observation x_t as a linear combination of the p previous observations

$$\hat{x}_t = \frac{1}{p}x_{t-1} + \frac{1}{p}x_{t-2} + \dots + \frac{1}{p}x_{t-p}. \quad (6)$$

In the interest of simplicity, in this paper we fix $p = 2$. That is, we calculate the moving average of an observation x_t as a function only of the observations x_{t-1} and x_{t-2} . We make the further assumption that the weight afforded to x_{t-1} and x_{t-2} is uniform (i.e. the estimate for x_t is the simple arithmetic mean of the prior two observations). This gives us the moving average:

$$\hat{x}_t = \frac{1}{2}x_{t-1} + \frac{1}{2}x_{t-2}. \quad (7)$$

Clearly we could expand our model in several ways. We could increase k , leading to a smoother model. We could also follow the intuition that our k prior observations should not have equal say in estimating x_t ; perhaps more recent observations should count more than older ones. Sophisticated moving average calculations abound in the time series literature. However, in this paper the simple model of Eq. 7 will suffice. In fact the crux of our argument rests on the simplicity of this model. The premise of this paper is that powerful discriminators show complex behavior over time; a simple model fits them poorly. On the other hand, words with negligible value for IR will show regularity over time that is simple enough to fit closely with a model such as Eq. 7.

Before proceeding, it bears repeating that the moving average described in this section is a simple smoothing operation. We do not use the term *moving average* in the sense used by Box et al. (1994). However, we do turn to the Box-Jenkins approach later in this section.

4.2 Simple Linear Regression

The form of the weakly autoregressive model in Eq. 5 is the same as a classic linear regression. Using a simple linear regression affords us a second time series model.

Let X^+ be the lag-1 differenced version of the time series X with a zero prepended to it. Thus for all integers $t \in 1 \dots (n-1)$ we have $x_{t+1}^+ = x_t$. This setup gives the simple linear regression model:

$$x_t = \beta_0 + \beta_1 x_t^+ + \epsilon \quad (8)$$

where ϵ is the iid noise variable described above and β_0 and β_1 are the model parameters.

During classical regression the parameters β_0 and β_1 are estimated using the method of least-squares, as described in Neter et al. (1996), *inter alia*. The least-squares estimates of these parameters are

$$\hat{\beta}_1 = \frac{Cov(x, x^+)}{Var(x^+)} \quad (9)$$

$$\hat{\beta}_0 = \bar{x} - \hat{\beta}_1 \bar{x}^+ \quad (10)$$

where \bar{x} and \bar{x}^+ are the sample means of X and X^+ , respectively.

Unlike the scenario outlined here, in many simple regression settings the independent variable is non-stochastic. That is, the regressors are assumed to be fixed. However, the model in Eq. 8 uses the prior observation x_{t-1} to estimate the current observation x_t . This departure from typical regression models is discussed by Anderson (1994) who shows that the effect of relying on stochastic regressors is that the least-squares estimators are not guaranteed to equal the maximum likelihood estimators. However, Anderson shows that the least-squares and maximum-likelihood estimators are typically close under these circumstances, and he demonstrates the utility of this type of regression for time series analysis.

4.3 Box-Jenkins Autoregressive Models

The Box-Jenkins approach to modeling (under the umbrella term autoregressive moving average, or ARMA, models) provides a formal autoregressive model for a time series X (Box and Pierce, 1970).

As in the previous discussion, a Box-Jenkins autoregressive model estimates the value of a time series at time t as a function of the values of the series at $x_{t-1}, x_{t-2}, \dots, x_{t-p}$. An autoregressive model of order p takes the form:

$$x_t = \mu + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \dots + \phi_p x_{t-p} + \epsilon_t \quad (11)$$

where the time series X is stationary, $\phi_1, \phi_2, \dots, \phi_p$ are the model parameters and ϵ_t is iid with constant variance σ^2 . We refer to the model given in Eq. 11 as $AR(p)$. Throughout this paper we deal with the simplest autoregressive model $AR(1)$:

$$x_t = \mu + \phi x_{t-1} + \epsilon_t. \quad (12)$$

That is, we assume that the observation at time t is a linear function of the value of the series at time $t-1$. Because the autoregressive model assumes stationarity, when applying autoregressive models below, we will always work with time series that have been differenced with lag 1.

Following Shumway and Stoffer (2006, Eqs. 3.103-3.104) we derive the conditional maximum likelihood estimator for the Box-Jenkins autoregressive model:

$$\hat{\phi} = \frac{\sum_{t=2}^n (x_t - \bar{x}_{(2)})(x_{t-1} - \bar{x}_{(1)})}{\sum_{t=2}^n (x_{t-1} - \bar{x}_{(1)})^2} \quad (13)$$

$$\hat{\mu} = \frac{\bar{x}_{(2)} - \hat{\phi} \bar{x}_{(1)}}{1 - \hat{\phi}} \quad (14)$$

where $\bar{x}_{(1)} = (n-1)^{-1} \sum_{t=1}^{n-1} x_t$ and $\bar{x}_{(2)} = (n-1)^{-1} \sum_{t=2}^n x_t$. The stochastic nature of the regressors leads to the estimators shown here. That is, because the regressors are themselves random, the conditional least-squares method of Eq. 13 is used for fitting $AR(1)$ under the Box-Jenkins approach³.

³In the experiments described below, we used the `arima` function from the R programming environment (<http://www.r-project.org>) to estimate $AR(1)$.

4.4 Summary

In this section we have described three methods of modeling time series data. These methods are summarized in Table 1. Each of these methods will yield a corresponding method of term weighting in Section 5.

Table 1: Summary of time series models proposed in this paper. For each model we list its name, the equation that defines it, the specific data that it is used with, and a brief description.

| Model | Equation | Data | Description |
|-------------------|----------|------------------------------------|---------------------------------------------------------------------------------|
| moving average | 7 | unit-normalized, non-differenced | x_t is the arithmetic mean of x_{t-1} and x_{t-2} . |
| linear regression | 8 | unit-normalized, non-differenced | x_t is linear on x_{t-1} , using least-squares estimators. |
| autoregression | 12 | unit-normalized, lag-1 differenced | x_t is linear on x_{t-1} , using conditional maximum likelihood estimators. |

5 Term Weights from Time Series Analysis

Our argument that term weight should be inversely proportional to linear model goodness of fit clearly presupposes a metric for gauging goodness of fit. The model building and model selection literature is very mature, and this section relies only on its most expedient results. Readers may find deeper treatment in (Burnham and Anderson, 2002).

As a preface, our measure of each model’s goodness of fit is its root mean squared error (RMSE). That is, the global weight of each term in the vocabulary (comparable to its IDF) is the RMSE of a model—moving average, linear regression, or autoregression—fitted to its corresponding time series. The remainder of this section defines RMSE and offers our motivation for using it.

For the family of linear models, assessing goodness of fit typically begins with the fitted model’s residuals. Let $\hat{y} = f(x)$ be a model that estimates a dependent variable y based on the independent variable x . Given this model and a set of n data points represented as n -vectors \mathbf{x} and \mathbf{y} with the fitted values $\hat{\mathbf{y}}$, the model residuals are simply $\hat{\mathbf{y}} - \mathbf{y}$.

Assessing goodness of fit often considers model residuals in the form of the residual sum of squares (RSS):

$$RSS = \sum_{i=1}^n (\hat{y}_i - y_i)^2. \quad (15)$$

In our case, $f(x)$ could be the moving average, linear regression, or autoregression described above. But in all cases, the residual sum of squares is small when the fitted values closely approximate the observed values of the dependent variable.

From Eq. 15 the so-called mean squared error (MSE) follows naturally:

$$MSE = \frac{RSS}{n - 2} \quad (16)$$

where the denominator reflects the fact that our models lose two degrees of freedom due to our having estimated two parameters. If we make the assumption that the error terms in our model are distributed

$N(0, \sigma^2)$, then the MSE is an unbiased estimator of σ^2 , the model variance. Likewise, an unbiased estimator of the standard deviation is simply the so-called root mean squared error:

$$RMSE = \sqrt{MSE}, \tag{17}$$

the positive square root of the MSE.

More explicitly, let X be the length- n time series observed for the term x . Let \mathcal{M} be a model of this time series. In this paper we consider three types of models—moving average, linear regression, and autoregression. The global term weight for x given \mathcal{M} is:

$$weight_{\mathcal{M}}(x) = \sqrt{\frac{\sum_{i=1}^n (\hat{x}_i - x_i)^2}{n - 2}} \tag{18}$$

where \hat{x}_i is the model’s estimate of x_i . However, the second order moving average operator returns a time series of length $n - 2$. For MA, then, we sum over $n - 2$ observations.

In the case of our regression models—linear regression and autoregression—it is not unreasonable to make the assumption of Gaussian error terms. However, the moving average model is non-parametric. Thus the use of RMSE for our moving average models is heuristic. Nonetheless, for all three models, the intuition is the same: terms in which a model in the form of Eq. 5 yields a high RMSE exhibit behavior over time that is more complex than terms with low-RMSE models. We argue that strong discriminators’ collection frequency at time t is difficult to predict given observations at prior times. This difficulty translates to poor model fit as measured by RMSE. Thus we weight each term by the RMSE of the model—moving average, linear regression, or autoregression—of its time series.

6 Experimental Evaluation

To assess the usefulness of the time-based term weights proposed in this paper we conducted a series of Cranfield-style experiments using data from several tasks undertaken at TREC. This section describes these experiments as well as their results.

6.1 Data Used for Experimentation

We used four test collections in the following experiments. Table 2 summarizes these collections. The LA Times data are from Disk 4 of the TREC Tipster collection. The acquaint corpus was used with fifty topics that were gathered in support of the TREC robust track in 2004⁴. These topics (drawn from 301-450) were selected due to their demonstrated difficulty in the 2003 robust track. The genomics data consists of HTML documents representing full-text scientific articles. The documents in wt10g are also full-text HTML, but these were crawled from the Web at large.

The rightmost column of Table 2 describes the temporal nature of each collection. By “temporal nature” we mean the way that the collection was divided into intervals to allow time series analysis. The two news collections—LA Times and acquaint—are inherently sequential; each file supplied by NIST contains articles published on a particular day. To create time series, then, we simply divided each of these collections into week-long intervals. This lead to a length of $n = 104$ observations for the LA Times data and $n = 160$ for acquaint. The genomics and wt10g fall less obviously into discrete time intervals. While it is true that each

⁴see <http://trec.nist.gov/data/robust/04.guidelines.html>

Table 2: Test Collections Used for Experimentation

| Corpus | # Docs. | Topics | Doc. Type | Temporality |
|----------|-----------|-----------------------|---------------------|-------------|
| LA Times | 131,896 | 401-450 (50) | News | Weekly |
| aquaint | 1,033,461 | robust 2004 (50) | News | Weekly |
| genomics | 162,259 | 160-187, 200-235 (64) | Scientific articles | Artificial |
| wt10g | 1,692,096 | 501-550 (50) | Web | Artificial |

document in these collections has a time stamp, we simply created “proxy” intervals for these collections by dividing them into slices of roughly equal size.

The genomics data are stored in individual HTML files—one file per indexable document. To build the genomics data time series we simply generated a list of these documents using the unix `find` command and created proxy intervals by taking contiguous groups of $n = 500$ documents from the resulting list. This led to the genomics data time series being of length $n = 163$.

Temporal units for wt10g were derived by concatenating five of the collection’s text files. The wt10g text files each contain several hundred indexable documents. The series for wt10g had $n = 1033$.

For all queries except those in the genomics collection, two query representations were run. The *short* queries described below contain only each topic’s TITLE field. So-called *long* queries used the TITLE and DESCRIPTION fields. The genomics queries were available in only one representation, which was analogous to the *long* queries (mean topic length was 10.22 words).

All indexes were built using no stoplists and no stemming. While stoplists and stemming might improve performance, we omitted such pre-processing in efforts to test the proposed term weighting methods rigorously (i.e. a useful term weighting model should mitigate the influence of stopwords automatically).

6.2 Baseline retrieval methods

The experiments reported here test retrieval using the time-based term weights described in Section 5 against two baseline retrieval algorithms. All retrieval systems (the time-based runs and the baselines) were implemented using the open-source Lemur toolkit⁵. The baselines were:

1. the Kullback-Leibler (KL) divergence model described in (Zhai and Lafferty, 2006).
2. Okapi weighting using BM25 weights as in (Robertson et al., 1995)

The baseline KL model used Dirichlet smoothing with hyperparameter $\mu = 1000$ (cf. (Zhai and Lafferty, 2004)). The BM25 method used the following parameters:

- $k1 = 0.75$
- $b = 0.55$

⁵<http://lemurproject.org>

- $k3 = 7$
- $q = 0.5$.

These are similar to the default parameters given in Lemur, and were chosen empirically to give the highest performance with respect to mean average precision (MAP) on a separate training corpus. Thus the BM25 baseline system’s retrieval status value for a document D with respect to a query Q is:

$$RSV(D, Q) = \sum_{i \in Q} \log \frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} \cdot \frac{(k_1 + 1)f_i}{f_i + k_1 \cdot (1 - b + b \cdot \frac{|D|}{|D_{avg}|})} \cdot \frac{(k3 + 1)qf_i}{k3 + qf_i} \quad (19)$$

where N is the number of documents in the corpus, $n(q_i)$ is the document frequency of term q_i and f_i is the number of times term q_i occurs in D . Finally $|D|$ is the length of document D and $|D_{avg}|$ is the average document length in the collection. The term qf_i is the frequency of term q_i in the query. The summation is taken over all words in the query.

It is worth noting that the leftmost factor in Eq. 19 is a variant of the IDF for term q_i .

6.3 Experimental System

To measure the effectiveness of the time-based weighting methods described in Section 5 we implemented systems using the following scoring formula. For a query Q and a document D the document score is

$$score(D, Q) = \sum_{i \in Q} w_{\mathcal{M}} \cdot \frac{(k_1 + 1)f_i}{f_i + k_1 \cdot (1 - b + b \cdot \frac{|D|}{|D_{avg}|})} \cdot \frac{(k3 + 1)qf_i}{k3 + qf_i} \quad (20)$$

where $w_{\mathcal{M}}$ is one of the time-based term weights described above in Section 5. Thus the experimental system differs only from the Okapi baseline in the global term weight used in calculating query-document scores. Due to this similarity, Okapi and the experimental systems are directly comparable with respect to the global term weighting factor. This is in contrast to the KL baseline, whose document and query modeling differ from Okapi and the experimental systems. Thus in the results below, statistical comparisons are only made with respect to Okapi.

6.4 Experimental Results

The dependent variable we aimed to test in these experiments was retrieval effectiveness. We operationalized this using two statistics, mean average precision (MAP) and R-precision (Rprec). MAP affords a useful summary of performance over a broad range recall levels. We also report R-precision, which gives insight into the behavior of a retrieval model at relatively high levels of precision. We report R-precision instead of metrics such as P@10 because of idiosyncrasies in several of the test collections used in our experiments. R-precision is the precision observed at k documents retrieved, where k is the total number of documents relevant to the query. Unlike P@10, Rprec is well suited to situations where the queries in a test collection have widely varied numbers of relevant documents. This is the case in both the genomics data and the Web data.

Tables 3 and 4 summarize our experimental results. The tables show MAP and Rprec averaged over all queries for each data set. The topmost row of data in each table gives results for BM25, our baseline

Table 3: Mean average precision (MAP) results. Numbers in parentheses give the percent improvement over the Okapi baseline run. A † indicates that an improvement was statistically significant ($p < 0.05$) on a one-sided, paired t -test.

| | LA | | Aquaint | | Genomics | wt10g | |
|-------|-------------------|-------------------|-------------------|---------------------|---------------------|-------------------|---------------------|
| | short | long | short | long | | short | long |
| Okapi | 0.1994 | 0.1895 | 0.1719 | 0.1624 | 0.2325 | 0.1610 | 0.1645 |
| KL | 0.1995 | 0.2023 | 0.1761 | 0.1789 | 0.2084 | 0.1538 | 0.1625 |
| MA | 0.2029 (+1.76) | 0.2033 (+7.28) | 0.1744 (+1.54) | 0.1748 (+7.64)† | 0.2870 (+23.75)† | 0.1614 (+0.25) | 0.1949 (+18.48)† |
| LR | 0.2036 (+2.11) | 0.2031 (+7.18) | 0.1746 (+1.57) | 0.1700 (+4.68) | 0.2486 (+6.92) | 0.1638 (+1.74) | 0.1902 (+15.62)† |
| AR | 0.2034 (+2.00) | 0.2035 (+7.39) | 0.1720 (+0.00) | 0.1802 (+10.96)† | 0.2529 (+8.88)† | 0.1634 (+1.49) | 0.1921 (+16.78)† |

Table 4: R-precision (rprec) results. Numbers in parentheses give the percent improvement over the Okapi baseline run. A † indicates that an improvement was statistically significant ($p < 0.05$) on a one-sided, paired t -test.

| | LA | | Aquaint | | Genomics | wt10g | |
|-------|--------------------|-------------------|--------------------|-------------------|---------------------|-------------------|-------------------|
| | short | long | short | long | | short | long |
| Okapi | 0.2155 | 0.2034 | 0.2287 | 0.2261 | 0.2577 | 0.1875 | 0.1959 |
| KL | 0.2112 | 0.2115 | 0.2371 | 0.2424 | 0.2436 | 0.1891 | 0.2070 |
| MA | 0.2281 (+5.85)† | 0.2216 (+8.95) | 0.2399 (+4.90)† | 0.2324 (+2.79) | 0.3002 (+16.49)† | 0.1883 (+0.43) | 0.2083 (+6.33) |
| LR | 0.2241 (+3.99)† | 0.2219 (+9.10) | 0.2415 (+5.60)† | 0.2268 (+0.31) | 0.2799 (+8.61)† | 0.1950 (+4.00) | 0.2064 (+5.36) |
| AR | 0.2251 (+4.46) | 0.2229 (9.59) | 0.2395 (+4.72)† | 0.2300 (+1.72) | 0.2860 (+10.99)† | 0.1949 (+3.95) | 0.2078 (+6.07) |

system. For purposes of comparison, the second row shows results obtained using the KL divergence model. Remaining rows give performance statistics for each of the three proposed time-based weighting methods using the retrieval model given in Eq. 20.

Beneath each experimental run’s data, Tables 3 and 4 show the percent of change with respect to the Okapi baseline. Cells marked with a † indicate that the corresponding change was statistically significant: $p < 0.05$ on a one-sided, paired t -test as recommended in Smucker et al. (2007).

All performance improvements are reported with respect to the Okapi baseline system. We report results using the KL model simply as an additional point of comparison. The KL approach, along with BM25 is considered state-of-the-art in contemporary IR, and thus we report its success. However, the comparison between Okapi and our test systems is direct; our systems differs from the Okapi baseline only in the IDF-like global term weights they use.

The data in Tables 3 and 4 suggest that the time-based approaches detailed in this paper are promising. All time-based runs gave MAP and Rprec greater than Okapi. While it is the case that time-based weighting’s improvements over Okapi were not always statistically significant (and in some cases, the improvements were small indeed) 15 runs did deliver statistically significant improvements when we used RMSE of a time series model for term weighting.

Tables 3 and 4 show an interesting result with respect to improvements in MAP and Rprec as they relate to query length. In Table 3 we see that the strongest improvements in MAP came with longer queries. On the other hand in Table 4, we see improvements in Rprec primarily with short queries. This result suggests that time-based term weights are effective in two very different settings. That is, they appear to work well for terse queries at high levels of precision as is common in Web search. On the other hand, they also gave strong results for more expressive queries, for which we might presume users would be invested enough to view more search results.

Certainly the strongest results for time-based weighting came in the case of the genomics data. This data set is different from our other collections in a few respects. Most obviously, the genre of its documents is unique. Because it consists of scholarly articles in a narrow topical domain, the genomics corpus contains a highly technical vocabulary. This fact is also reflected in the genomics queries. These queries are highly specific, requesting information on particular genes, etc. But they express these information needs verbosely.

We attribute the success of time-based weights on the genomics data to the possibility that word frequency and discriminatory power are related in a manner more complicated than weights based on rarity alone can account for. To guide our discussion of this hypothesis, we present Figures 3 and 4. The panels in each figure plot a particular term’s weight in decreasing order of IDF. The points in Figure 3 are the terms in the long queries 401-450, with weights calculated on the LA Times data. Figure 4 shows weights for the terms from the genomics data, calculated from the genomics corpus. The y -axis for the time-based weights show the log of the RMSE because raw RMSE’s distribution had an extremely long tail and thus did not graph well.

In Figures 3 and 4 we see that the distribution of weight over terms differs dramatically between the LA Times data and the genomics data where the time-based weights performed especially well. The stark divergence between IDF and the time-based approaches on the genomics data suggests that behavior over time yields valuable information in the context of a highly technical IR environment such as the genomics test collection. We speculate that in the genomics collection, many terms are rare and thus see high IDF. But for an “important” rare term, we would expect to see spikes in frequency when an article is added to the collection that mentions the term often. That is, a term may be rare, accumulating its few occurrences essentially at random. On the other hand a rare term may achieve its occurrences in bursts when an individual document that treats the topic that it refers to makes its way into the index. IDF would not

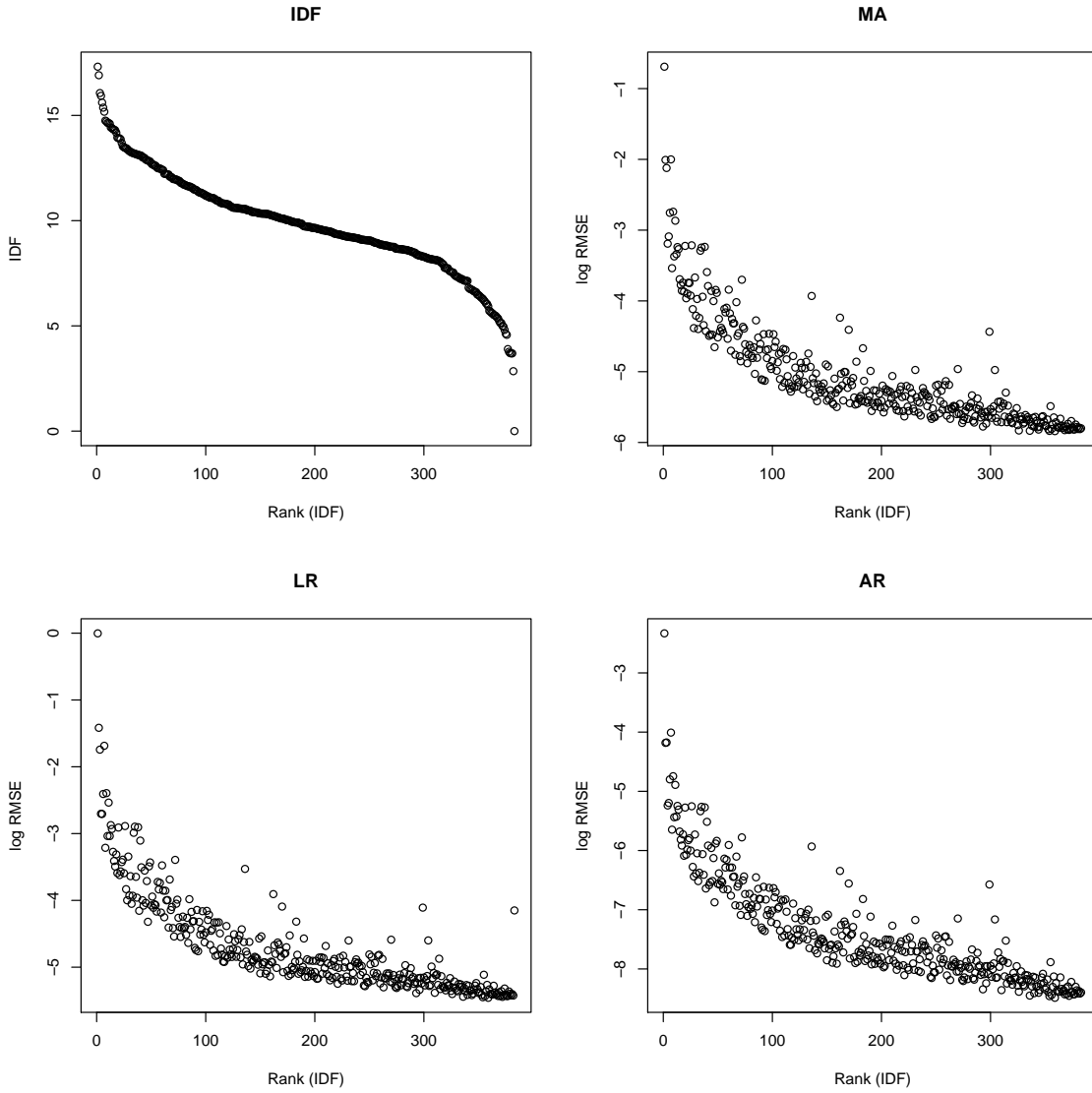


Figure 3: Term weights for words in long TREC topics 401-450, listed in decreasing order of IDF on the LA Times data.

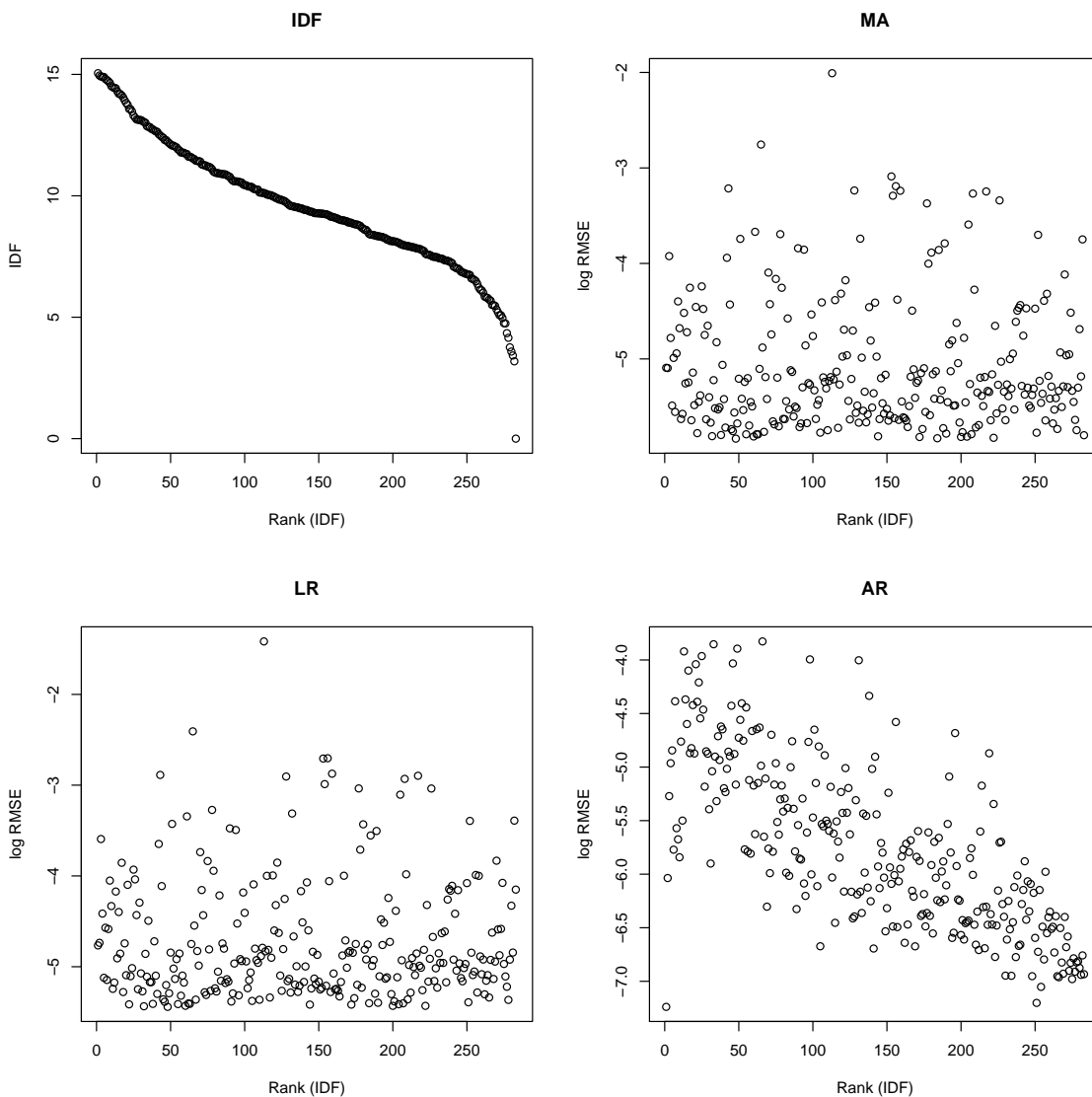


Figure 4: Term weights for words in TREC genomics collections, listed in decreasing order of IDF on the genomics data.

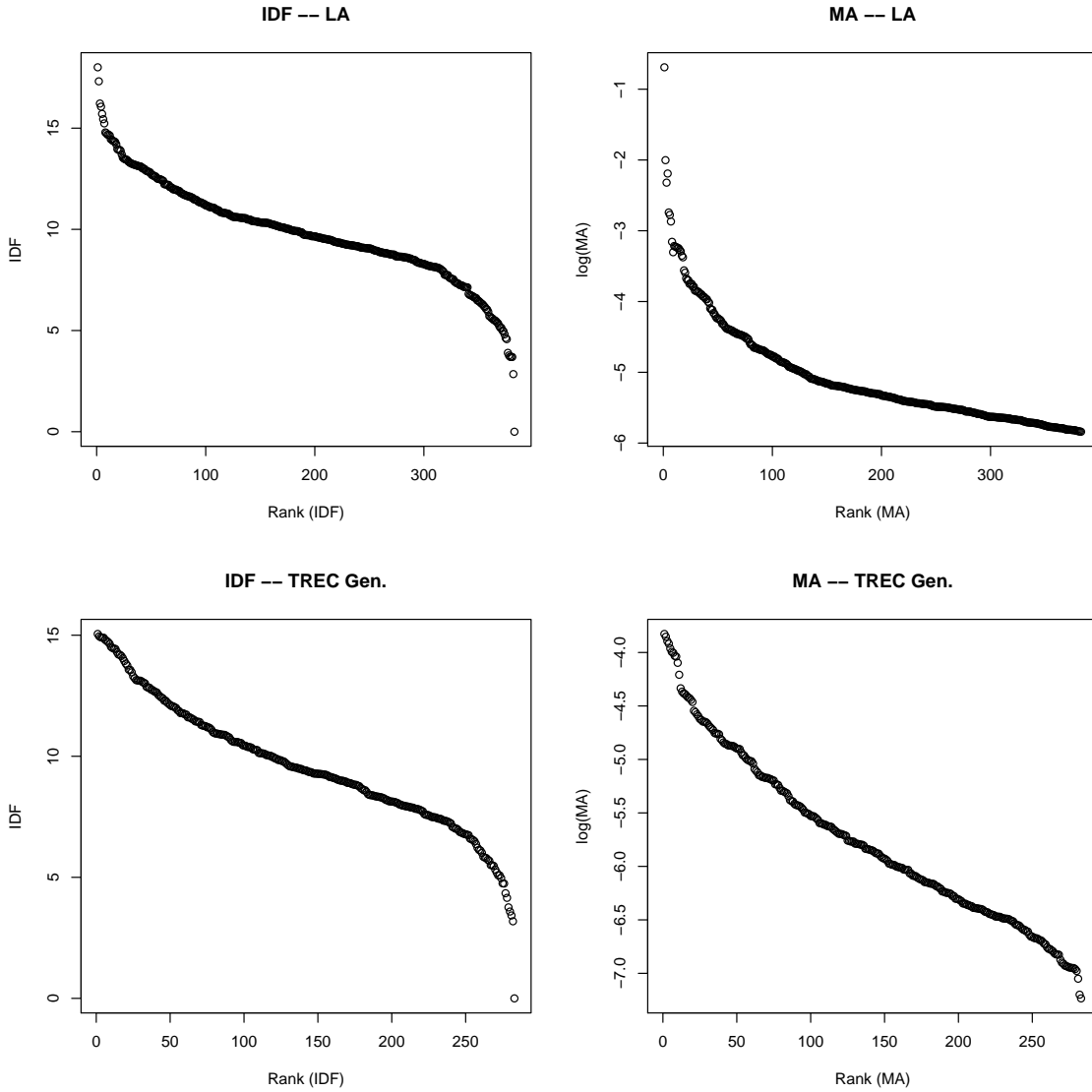


Figure 5: Distribution of IDF and MA-derived RMSE (on a log scale) for the LA Times and Genomics data.

distinguish between these cases, while our time-based measures would. This distinction would explain the difference between the Okapi results and time-based performance.

Figures 3 and 4 also speak to results that are evident in Tables 3 and 4. First, it is the case that RMSE from term time series are measuring phenomena that are not identical to IDF. That is, our time-based approach does not weight terms in the same way as IDF but through a different avenue. Instead, we see that the distribution of log-RMSE is concave, following a much different distribution than IDF as is evident in Figure 5.

Also Figures 3 and 4 show that the differences between the three proposed time-based weights are negligible. None of the time-based weights definitively outperformed the other two approaches. Likewise, with the exception of AR’s distribution on Figure 4 the distribution of all three log RMSEs is nearly identical.

Table 5: MAP calculated from time series of differing lengths. Column headings refer to the proportion of the original time series observations retained for each run. MAP results were obtained using *long* queries.

| Method | Full | 1/2 | 1/4 | 1/6 | 1/8 | 1/12 |
|----------|--------|--------|--------|--------|--------|--------|
| LA Times | | | | | | |
| MA | 0.2029 | 0.2054 | 0.2049 | 0.2022 | 0.2015 | 0.1878 |
| LR | 0.2036 | 0.2052 | 0.2000 | 0.1940 | 0.1842 | 0.1739 |
| AR | 0.2034 | 0.1747 | 0.1721 | 0.1697 | 0.1692 | 0.1689 |
| Aquaint | | | | | | |
| MA | 0.1748 | 0.1744 | 0.1724 | 0.1699 | 0.1680 | 0.1687 |
| LR | 0.1700 | 0.1704 | 0.1687 | 0.1724 | 0.1662 | 0.1652 |
| AR | 0.1802 | 0.1747 | 0.1721 | 0.1697 | 0.1692 | 0.1689 |
| Genomics | | | | | | |
| MA | 0.2870 | 0.2568 | 0.2568 | 0.2466 | 0.2466 | 0.2464 |
| LR | 0.2486 | 0.2462 | 0.2462 | 0.2419 | 0.2419 | 0.2347 |
| AR | 0.2529 | 0.2261 | 0.2261 | 0.2398 | 0.2398 | 0.2378 |
| wt10g | | | | | | |
| MA | 0.1949 | 0.1916 | 0.1902 | 0.1797 | 0.1742 | 0.1739 |
| LR | 0.1902 | 0.1813 | 0.1774 | 0.1728 | 0.1683 | 0.2347 |
| AR | 0.1921 | 0.1837 | 0.1811 | 0.1749 | 0.1720 | 0.1701 |

Table 5 speaks to the matter of how frequently we make observations when constructing time series for a corpus. Earlier we discussed the methods that we followed to divide the four TREC test collections shown in Table 2 into time series. Of course other approaches could have guided us. For instance we could have sampled the news corpora monthly (or daily) instead of weekly. Likewise, each “observation” on the genomics data could have consisted of 100 (instead of 500) documents. How does defining the unit of time bear on the approaches detailed in this paper?

Table 5 reports MAP calculated using our time-based term weights when differing temporal windows are used. The column labeled *Full* gives MAP for the test collection as reported above as a baseline. All MAP scores were derived using long queries. The following columns give results from increasingly widened windows. Thus results in the column labeled 1/2 come from term weights calculated at every other observation in the original data (i.e. the series was constructed using two-week intervals for the news collections). The 1/4 column uses every fourth observation. In the case of the news data, this amounts to using a monthly window instead of the weekly window we reported above. The remaining columns use one sixth, one eighth, and one twelfth of the original observations.

Table 5 suggests that widening the sampling window does degrade performance. The actual decline in performance is small for the most part if we read from left to right in the table. However, there is a

systematic trend for widened windows to yield lower MAP. This is not surprising; the 1/12 sampled acquaint time series are only of length $n = 14$, providing very little data to fit a time series model.

Table 5 does not suggest that any one of the three time-based approaches is more robust than the others. Across the four test collections, different models withstand the widened observation window differently. Perhaps the most dramatic failure, however, lies in the case of MA on the genomics data. Here we see a large drop in MAP as we shorten the time series. However, before labeling MA as especially fragile, it is worth noting that MA’s performance on the genomics data was far better than the others; thus its performance at the “full” time window appears to have been an outlier.

To summarize our results, this paper specified three types of “weakly autoregressive” models: moving average, linear regression, and autoregression. In all of these cases the model predicts the value of a time series X at time t as a linear function of prior observations on X . The term weights we proposed follow from these models. They are simply the root mean squared error of each model.

The experimental results reported this section are promising. Time-based approaches yielded statistically significant improvements over a state-of-the-art baseline model on several runs, and yielded at least marginal improvements on all tested conditions. While the methods proposed here showed improvements over baseline retrieval models, we found no compelling difference among the three tested time series analysis methods. All three methods showed similar (though not identical) performance, and they all appeared equally robust with respect to the matter of time series window size.

7 Discussion and Conclusion

Most corpora change over time. We have argued that this change is informative with respect to IR. Specifically, the premise of this paper is that the collection frequency of terms with strong topical discriminatory power changes over time such that these terms’ collection frequencies are not weakly autoregressive. If we approximate terms’ time series using the linear techniques presented here, we will see poor model fit to terms with unpredictable distributions. We argue that weighting terms to an extent that is inversely proportional to the goodness of fit of a linear model induced on their time series provides a useful basis for term weighting in IR.

This argument begs several questions. Why should strong discriminators show non-autoregressive time series? What causes poor discriminators’ collection frequency at time t to be predictable by their frequency at time $t-1$? As we seen in Figure 1 the lion’s share of the variance for the term *the* depends on the collection size. When we de-trend the series via differencing in Figure 2 the time series for *the* is nearly flat. On the other hand, movement in the series for *schengen* is less dependent on collection size; the differenced series retains relatively high variance. In essence, a weak discriminator will make up roughly the same proportion of the overall vocabulary at time t as it did at time $t-1$. A strong discriminator’s frequency in a particular sample of text (i.e. x_t) is highly random.

This argument shows that considering time in term weighting is not orthogonal to traditional frequency-based weights. Measures such as IDF have been shown to have probabilistic and information-theoretic interpretations (Church and Gale, 1995). This paper invites using an information-theoretic lens to understand IR time series. Church and Gale (1995) argue that a term’s IDF and its entropy are closely related. In this and prior work, the notion of entropy has been brought to bear on a static corpus. That is, a word’s probability distribution is estimated by its pattern of occurrence in a corpus at a given time. This paper extends this idea. We argue that analyzing a term’s behavior during the lifespan of a corpus gives useful information about that term. In future work we plan to couch this argument in probabilistic terms in order to show more clearly how the time series analysis advanced here relates to prior work relating term weighting

to information theory.

Readers may note that we have only discussed growth in corpora, assuming that change is limited to the addition of documents when in fact real-world settings often see the deletion of documents from an index. We justify this omission on the grounds that a time series derived from a corpus lifespan that includes deleted documents may be re-ordered to approximate a lifespan of constant growth. That is, there is no reason (using the models proposed here) to take the observed ordering of a time series as gospel. Thus if C is the time series containing the total number of documents in a collection and $c_{t+1} < c_t$ we may simply swap these observations (and all corresponding terms' series), repeating this process as needed to render the time series as a non-decreasing series.

To expand on the research reported here we plan to pursue both theoretical and practical avenues. First, we suspect that combining temporal information with IDF-type information will lead to retrieval results that outpace performance obtained from each approach taken in isolation. This paper has treated matters of term rarity and temporality as distinct phenomena. However, we anticipate that bringing both types of analysis to the term weighting problem will improve performance.

Second, the methods introduced here invite a novel theoretical approach to IR. Time series analysis is often used in service to quantitative finance. In upcoming work we will present a model based on work in econometrics. Specifically, we shall consider each term to be an asset within a stock market. Consequently each document may be considered to be a portfolio of asset investments. In this scenario it is natural to rank documents by the wealth they return or can be expected to return⁶. We expect that this method will allow us to marshal the temporal data we have used in this paper in a unified theoretical framework.

References

- Aizawa, A. (2003). An information-theoretic perspective of tf-idf measures. *Information Processing and Management*, 39(1):45 – 65.
- Allan, J., editor (2002). *Introduction to topic detection and tracking*. Kluwer Academic Publishers, Norwell, MA, USA.
- Anderson, T. W. (1994). *The Statistical Analysis of Time Series*. Wiley-Interscience, New York.
- Box, G., Jenkins, G. M., and Reinsel, G. (1994). *Time Series Analysis: Forecasting and Control*. Prentice Hall, New York, 3rd edition.
- Box, G. and Pierce, D. (1970). Distribution of residual autocorrelation in autoregressive-integrated moving average time series model. *Journal of the American Statistical Association*, 65:1509–1526.
- Burnham, K. P. and Anderson, D. R. (2002). *Model Selection and Multimodel Inference*. Springer, New York, 2nd edition.
- Church, K. W. and Gale, W. A. (1995). Poisson mixtures. *Natural Language Engineering*, 1:163–190.
- Cokol, M. and Rodriguez-Esteban, R. (2008). Visualizing evolution and impact of biomedical fields. *J. of Biomedical Informatics*, 41(6):1050–1052.
- Diaz, F. and Jones, R. (2004). Using temporal profiles of queries for precision prediction. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 18–24, New York, NY, USA. ACM.

⁶A small amount of recent work has used financial models in IR (Wang and Zhu, 2009; Wang, 2009). However, this work operates under a different framework from what we propose, considering documents (as opposed to terms) as assets in the market.

- Dubin, D. (1999). Toward more robust discrimination-based indexing models. Technical Report UIUCLIS-1999/7+IRG, Graduate School of Library and Information Science, University of Illinois, Urbana-Champaign.
- Gruhl, D., Guha, R. V., Liben-Nowell, D., and Tomkins, A. (2004). Information diffusion through blogspace. In *WWW*, pages 491–501.
- Guralnik, V. and Srivastava, J. (1999). Event detection from time series data. In *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 33–42, New York, NY, USA. ACM.
- Hiemstra, D. (2000). A probabilistic justification of tf-idf term weighting. *International Journal of Digital Libraries*, 3(2):131–139.
- Jones, K. S. (1979). Search term relevance weighting given little relevance information. *Journal of Documentation*, 15:133–144.
- Jones, R. and Diaz, F. (2007). Temporal profiles of queries. *ACM Trans. Inf. Syst.*, 25(3):14.
- Kleinberg, J. (2002). Bursty and hierarchical structure in streams. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 91–101, New York, NY, USA. ACM.
- Kumar, R., Novak, J., Raghavan, P., and Tomkins, A. (2004). Structure and evolution of blogspace. *Commun. ACM*, 47(12):35–39.
- Lafferty, J. and Zhai, C. (2001). Document language models, query models, and risk minimization for information retrieval. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 111–119, New York, NY, USA. ACM.
- Lavrenko, V., Schmill, M., Lawrie, D., Ogilvie, P., Jensen, D., and Allan, J. (2000). Mining of concurrent text and time-series. In *KDD-2000 Workshop on Text Mining*.
- Leskovec, J., Backstrom, L., and Kleinberg, J. (2009). Meme-tracking and the dynamics of the news cycle. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 497–506, New York, NY, USA. ACM.
- Li, X. and Croft, W. B. (2003). Time-based language models. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 469–475, New York, NY, USA. ACM.
- Liebscher, R. and Belew, R. K. (2003). Lexical dynamics and conceptual change: Analyses and implications for information retrieval. *Cognitive Science Online*, 1:4657.
- Metzler, D., Jones, R., Peng, F., and Zhang, R. (2009). Improving search relevance for implicitly temporal queries. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 700–701, New York, NY, USA. ACM.
- Miller, N. E., Chung Wong, P., Brewster, M., and Foote, H. (1998). Topic islands—a wavelet-based text visualization system. In *VIS '98: Proceedings of the conference on Visualization '98*, pages 189–196, Los Alamitos, CA, USA. IEEE Computer Society Press.
- Montgomery, D. C., Jennings, C. L., and Kulahci, M. (2008). *Introduction to Time Series Analysis and Forecasting*. Wiley, New York.
- Neter, J., Kutner, M. H., Nachtsheim, C. J., and Wasserman, W. (1996). *Applied Linear Statistical Models*. Irwin, Chicago, 4th edition.
- Nguyen, H., Parikh, N., and Sundaresan, N. (2008). A software system for buzz-based recommendations. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1093–1096, New York, NY, USA. ACM.

- Robertson, S. E., Walker, S., Jones, S., Beaulieu, M. H., and Gatford, M. (1995). Okapi at TREC-3. In *Proceedings of TREC-3, the 3rd Text REtrieval Conference*, pages 109–127. NIST.
- Roelleke, T. (2003). A frequency-based and a poisson-based definition of the probability of being informative. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 227–234, New York, NY, USA. ACM.
- Roelleke, T. and Wang, J. (2008). Tf-idf uncovered: a study of theories and probabilities. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 435–442, New York, NY, USA. ACM.
- Salton, G. and Buckley, C. (1987). Term weighting approaches in automatic text retrieval. Technical report, Cornell University, Ithaca, NY, USA.
- Shumway, R. H. and Stoffer, D. S. (2006). *Time Series Analysis and its Applications, with R Examples*. Springer, New York, 2nd edition.
- Smucker, M. D., Allan, J., and Carterette, B. (2007). A comparison of statistical significance tests for information retrieval evaluation. In *CIKM*, pages 623–632.
- Swan, R. and Allan, J. (1999). Extracting significant time varying features from text. In *CIKM '99: Proceedings of the eighth international conference on Information and knowledge management*, pages 38–45, New York, NY, USA. ACM.
- Swan, R. and Allan, J. (2000). Automatic generation of overview timelines. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 49–56, New York, NY, USA. ACM.
- Tsay, R. S. (2005). *Analysis of Financial Time Series*. Wiley-Interscience, 2nd edition.
- Wang, J. (2009). Mean-variance analysis: A new document ranking theory in information retrieval. In *Proceedings of the European Conference on Information Retrieval (ECIR)*, pages 4–16.
- Wang, J. and Zhu, J. (2009). Portfolio theory of information retrieval. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 115–122, New York, NY, USA. ACM.
- Wang, X. and McCallum, A. (2006). Topics over time: a non-markov continuous-time model of topical trends. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 424–433, New York, NY, USA. ACM.
- Wang, X., Zhai, C., Hu, X., and Sproat, R. (2007). Mining correlated bursty topic patterns from coordinated text streams. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 784–793, New York, NY, USA. ACM.
- Wu, H. C., Luk, R. W. P., Wong, K. F., and Kwok, K. L. (2008). Interpreting tf-idf term weights as making relevance decisions. *ACM Trans. Inf. Syst.*, 26(3):1–37.
- Zhai, C. and Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems*, 2(2):179–214.
- Zhai, C. and Lafferty, J. (2006). A risk minimization framework for information retrieval. *Information Processing and Management*, 42(1):31–55.