Extending BM25 with Multiple Query Operators

Roi Blanco
Yahoo! Research
Barcelona, Spain
roi@yahoo-inc.com

Paolo Boldi
Dipartimento di Informatica
Università degli Studi, Milano, Italy
boldi@dsi.unimi.it

ABSTRACT

Traditional probabilistic relevance frameworks for informational retrieval refrain from taking positional information into account, due to the hurdles of developing a sound model while avoiding an explosion in the number of parameters. Nonetheless, the well-known BM25F extension of the successful Okapi ranking function can be seen as an embryonic attempt in that direction. In this paper, we proceed along the same line, defining the notion of *virtual region*: a virtual region is a part of the document that, like a BM25F-field, can provide a (larger or smaller, depending on a tunable weighting parameter) evidence of relevance of the document; differently from BM25F fields, though, virtual regions are generated implicitly by applying suitable (usually, but not necessarily, positional-aware) operators to the query. This technique fits nicely in the eliteness model behind BM25 and provides a principled explanation to BM25F; it specializes to BM25(F) for some trivial operators, but has a much more general appeal. Our experiments (both on standard collections, such as TREC, and on Web-like repertoires) show that the use of virtual regions is beneficial for retrieval effectiveness.

Categories and Subject Descriptors

H.3.3 [**Information Storage Systems**]: Information Retrieval Systems

General Terms

Performance, Experimentation, Ranking

Keywords

Query processing, ranking, query segmentation, BM25

1. INTRODUCTION

Modern information retrieval ranking functions, like the ones used in today's search engines, employ a large number of features derived from different sources of evidence: matches of query terms in documents, document query-independent quality measures, and click-through information among others. Prevalent among those signals, and central to most standard retrieval approaches such as BM25 and language models, is the term frequency (tf) information (i.e., the number of times a term appears in a document). Approaches derived from the probabilistic retrieval model are implemented as a summation of "weights" of the query terms that appear in the document, where the weight is essentially a normalized version of term frequency.

Traditional probabilistic relevance frameworks for informational retrieval [30] refrain from taking positional information into account, both because of the hurdles of developing a sound model while avoiding an explosion in the number of parameters and because positional information has been shown (somehow surprisingly) to have little effect on average [34]. Recently, though, it has been proved that considering sequences of terms that form *query concepts* is beneficial for retrieval [6]. Those extensions build up on the Markov Random Field retrieval model (MRF) and use a linear combination of different concepts and query-term scores in order to derive a final score for a document. Furthermore, the well-known BM25F extension of the successful Okapi ranking function (the latter of which we are aiming on building upon) can be seen as an embryonic attempt in the same direction: the basic idea there is that each document is made of regions (fields) and some fields may provide stronger evidence of relevance than others.

In this paper, we proceed along the same line, defining the notion of *virtual region*: a virtual region is a part of the document that, like a BM25F-field, can provide a different evidence of relevance of the document (the amount of evidence is, like in BM25F, expressed by a weight). Differently from BM25F fields, though, virtual regions are generated implicitly by applying suitable operators to the query: such operators may (and typically will) use positional information.

The techniques we propose can be seen as a two-stage ranking: in the first stage, a number of operators are applied to the incoming query to individuate virtual regions within the document; in the second stage, the regions are ranked much in the same way as with BM25F, using the weights attached to each operator. The idea is that there are "stricter" operators, that give a stronger evidence (e.g., "I want that at least three of the query terms appear in a span of at most ten words"), but are less likely to appear in document, and other that are "weaker" (e.g., "I want at least one of the query terms appearing somewhere", that is

SIGIR'12, August 12–16, 2012, Portland, Oregon, USA.

the standard bag-of-words requirement) and contribute to recall.

The operator-based technique that we propose fits nicely in the usual eliteness model behind BM25 and provides a principled explanation to BM25F; it specializes to BM25(F) for some trivial operators, but we believe it has a more general appeal.

Abstracting from the positional aspect, we can think of our method as an attempt to understand more deeply the user's intent underlying a query [2], following the increasing interest in extracting features and learning about the query itself. For example, all major search engines are able to detect entities in queries in order to shortcut the user to an appropriate vertical (e.g., for e-commerce or news), to trigger different visualization schemes or simply to help ranking better the results that are being produced. The basic idea is that of being able to pre-process a plain set of query terms that a user submitted to build a model of the query (possibly with the help of contextual information, e.g., about the query session and/or the user's profile). This model might simply contain spelling corrections, term annotations or may exhibit more sophisticated expansions, obtained through gazetteers, synonym dictionaries or query-logs, just to name a few.

In general, we observe that information retrieval is moving from a document-centric to a query/user-centric approach, and modern search engines are investing large amounts of research in building better, more comprehensive query models. The question that we address in this paper is whether it is possible to extend the classical probabilistic formulation so as to accommodate in a natural way these extended query models for enhancing ranking, in both Web search and more classical TREC-like retrieval settings.

Contribution.

Summing up, this paper aims at proposing an extension of the probabilistic retrieval framework [30] that accounts for the information coming from queries and documents. Our approach can be seen as a principled way of integrating query-document features into a BM25-based model [32], by extending the event space using a number of *operators* that derive from a query model [7]. For instance, both BM25 and BM25F [33] could be regarded as a special case of the method presented here. The framework operates in a general manner by means of a set of operators that are materialized using query-derived information; each operator determines a (possibly empty) virtual region within the document, that is treated as a (weighted) field; query-term frequencies in each virtual region are used to compute the final score of the document. We shall frame our technique as an extension of BM25(F) and describe how it can be implemented efficiently. Finally, we provide experimental evaluation of our approach showing that the usage of operators outperforms state-of-the-art ranking that use just matching of query terms and is especially helpful for *difficult* queries. The software implementing our technique and used for the experiments will be made available at `http://mg4j.dsi.unimi.it/`.

2. RELATED WORK

Some lines of research attempt at addressing the issue of adding semantic information to documents and queries [22]. The model that we present (which could indeed encode [22] as a special case) can be seen as a template for grounding different graphical model instances, in the spirit of Markov Logic Networks [18, 29], even though in this paper we make no attempt to generalize the learning procedure of the probabilities involved in the model, and the inference we perform is restricted to one particular formulation and combination. However, the very structure of our method allows one to extend it to different combinations and aggregation functionals over probability distributions. Robertson *et al.* [33] introduced BM25F, a variation of BM25 that is able to deal with matches of query terms in different fields of the document, boosting them differently. Our framework stems from the same fundamental notions of BM25F and BM25 (term eliteness, re-weighting of term matches) and it is able to extend/accommodate both.

There are several recent papers that deal with spans of terms in ranking. Svore et al. [35] show that introducing spans of terms as a further feature for machine learning to rank model gives improvements over BM25. Other authors have dealt with the issue of incorporating these spans of terms into the language model framework, the first one being the Markov Random Field (MRF) model of Metzler and Croft [26], extending the language modeling framework for information retrieval [27, 39] to handle term dependencies. Some other approaches [11] compute the aggregated distance of matches and add it to the BM25 score, or define a kernel-like distance [23, 24] that can be successfully used for ranking by plugging it into a language-model divergence between the query and document estimation. One remarkable model close to ours is that of Bendersky *et al.* [5], who weight different query concepts using a parameterized combination of diverse importance features: those concepts can be single query keyword, phrases matching in the document, or matches of keywords that span a window of a certain size. The amount of matching concepts in the document are later integrated into the MRF ranking model. In our case we focus on extending BM25 and not the language modeling framework, the core difference being in the way information is aggregated for each term during ranking.

Besides taking spans into account, it may be beneficial to adopt some additional query segmentation technique, trying to grasp which words in a query should appear in shorter spans (or even consecutively), as successfully attempted in [28, 19].

3. GOALS AND GUIDING EXAMPLES

As explained in the introduction, the basic approach of this paper is to extract, from a given query, a number of regions in the document using suitable operators. Alternatively, you can think that a given *input query* is refined in a number of different ways using some refinement operators, that may (and typically will) use positional information; virtual regions are then the regions matching each of the refined queries.

As a concrete example, consider the following two operators:

- Φ_1 requires that at least any two words in the query appear either consecutively or with an extra word between them;
- Φ_2 just requires that at least one of the query words appears.

One could think of them as query-refinement operators; for example, using the query language syntax of MG4J [9], Φ_1 applied to the query `young nice girl` gives rise to the query:[1]

```
(young nice)~3 OR (young girl)~3 OR (nice girl)~3.
```

When a document is considered against this query, all the areas where at least two of the three queried word appear either consecutively or one word apart are selected. This will determine a (possibly empty) sub-document, in which we can count the frequency of each of the three query words.

Operators Φ_2, instead, applied to the same query will produce

```
young OR nice OR girl;
```

and would just extract the occurrences of either of the three query words from the document.

Clearly, a large frequency in the virtual region determined by Φ_1 would be far more predictive of relevance than that determined by Φ_2 (the latter would amount to actually counting the usual term frequency in the whole document).

To gain some experimental support for this intuition we performed the following experiment: we considered the topics 701-850 from the TREC GOV2 collection, and built queries using the words in their title. To each query, we applied three operators: the plain `or` operator (corresponding to the usual bag-of-words interpretation of the query), the `2-and` operator (satisfied only by documents that contain at least any two of the query terms) and the `2-gram` operator (satisfied by documents that contain two terms consecutively).

Then, for each matching document, we computed the frequency of the query terms within the virtual region and we determined if the document was relevant or not; the fraction of relevant documents is plotted against term frequency. Figure 1 shows the results of the experiment, and provides two fundamental evidences: first of all, stricter operators (2-grams, for instance) provide for the same term frequency a larger probability of relevance, as expected; secondly, the behavior shows in all cases the well-known phenomenon of saturation — as the frequency increases, relevance also increases but at a slower and slower pace, giving rise to the typical sigmoid-shaped function.

Our approach is blind with respect to the operators being considered, which is part of its generality. In our experiments, among other operators, we employ a supervised phrase and entity detection algorithm and feed the different query chunks through the model in order to produce a document score.

Most previous works have addressed the combination of scores in a linear fashion: Bendersky *et al.* [5, 6] and Li *et al.* [20] focused on extensions of the language modeling framework. Linear combinations of features are able to bring increased performance; however, when taking into account evidence coming from the same source of information it is beneficial to understand the distributional properties of the signal the model has to deal with. In these cases, the information employed for ranking is always taken from the number of times one term or a sequence of terms matches a document. In contrast, if one was to incorporate other features, like query-independent document quality [17], or click-based information [1], a linear combination might be good enough, as long as the features integrated into the model are not correlated. Note however that (somehow surprisingly) even link-based features such as PageRank [10] and BM25 [32] turn out to be non-independent, given the presence of query terms in the anchor text of Web pages [17].

Not assuming feature independence is especially important when devising more complex ranking models that embody a large number of features, such as those employed in learned ranking functions [21]. In case of machine learning frameworks, this dependence is somehow captured by the complexity of learned functions, which in general might incorporate an over-engineering of features.

[1]In MG4J, the ~ operator restricts matches to a span of words of a given maximum length.

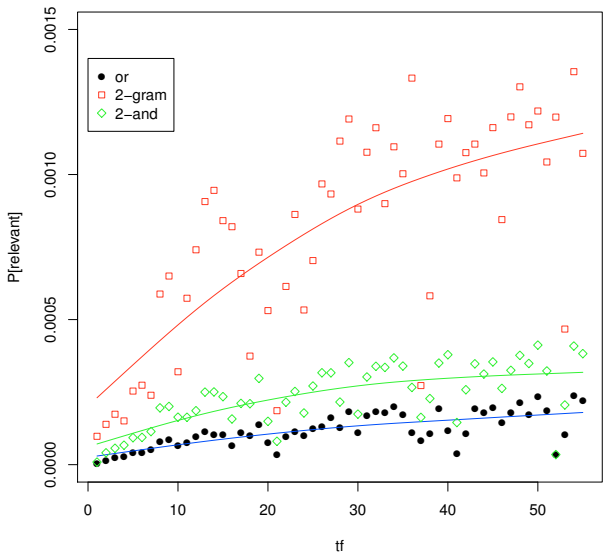

Figure 1: Term frequency versus probability of being relevant, depending on the operator applied to the query. Curves are obtained fitting the points through a spline with three degrees of freedom.

4. THE OPERATOR-BASED FRAMEWORK

Traditional probabilistic models, like BM25, assume that the relevance of a document to a query can be determined by aggregating individual contributions of the query terms. That is, given the binary random variable R representing relevance, and the vectors of random variables representing the document D and query Q, we want to rank documents according to their increasing odds-probability [30]. Here Q is (or can be thought of as) a set of terms, while D_t is a multi-state variable that encodes the features about the occurrence of term t in D (term frequency, position, etc.); we assume that those features contain a natural zero, corresponding to the absence of t and represented by $\mathbf{0}$. We let d and q denote two actual realizations of D and Q, and r ($\overline{r}$) represent the event $R = 1$ ($R = 0$, respectively), i.e., the document being relevant (irrelevant, respectively). Then, within the probabilistic framework, documents are ranked

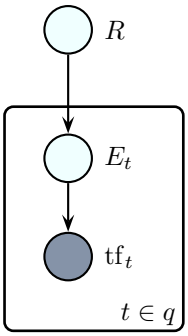

Figure 2: BM25 plate diagram representing the assumptions over variable independence

according to $p(r \mid Q = q, D = d)$, or equivalently

$$\begin{aligned}
\frac{p(r \mid Q = q, D = d)}{p(\overline{r} \mid Q = q, D = d)} &\propto_q \frac{p(D = d \mid r, Q = q)}{p(D = d \mid \overline{r}, Q = q)} \\
&\propto_q \prod_{t \in q} \frac{p(D_t = d_t \mid r)}{p(D_t = d_t \mid \overline{r})} \\
&\propto_q \sum_{t \in q, d_t > 0} \log \frac{p(D_t = d_t \mid r) \cdot p(D_t = \mathbf{0} \mid \overline{r})}{p(D_t = d_t \mid \overline{r}) \cdot p(D_t = \mathbf{0} \mid r)} \\
&\propto_q \sum_{t \in q, d_t > 0} w_{t,d},
\end{aligned}$$

where $w_{t,d}$ is the weight assigned for term t in document d, and you can think of d_t as being the term frequency of t in d, later on denoted by $\mathsf{tf}_{t,d}$.

In the derivation above, we assumed the terms to be *conditionally independent*, that is, $p(D_t = x, D_{t'} = y \mid r, Q) = p(D_t = d_t \mid r, Q) \cdot p(D_{t'} \mid r, Q)$ and $p(D_t = x, D_{t'} = y|\overline{r}, Q) = p(D_t = d_t|\overline{r}, Q) \cdot p(D_{t'}|\overline{r}, Q)$ for any pair of terms t and t'; this is a weaker assumption than term independence, in that we only require terms to be independent for each fixed query and relevance; this assumption is fundamental in practice, because it leads to tractable models, but it also has a deeper justification: as [16] proved, conditional term independence can be obtained when, for any given query, terms are statistically correlated but the correlation is the same on relevant and on non-relevant documents. Empirical evidence on retrieval performances of BM25 suggests that this indeed is often the case. For example, it is true that query terms `New` and `York` are correlated in relevant documents for the query `New York pizza`, but they are also correlated in the whole collection.

Further, the derivation imposes a vague prior assumption over terms not appearing in the query ($p(D_t = x \mid r) = p(D_t = x \mid \overline{r})$ if $t \notin Q$). This can be weakened in the case of query expansion by explicitly linking unseen query terms to relevance.

The final arithmetic trick in the above derivation, known as *removing the zeroes*, is used to eliminate from the final score calculation terms that are not present in the document.

The determination of term weights in BM25 is based on the assumption that there is an *Elite* random variable, which can be cast as a simple topical model that perturbs the distribution of words over the text. That is [30], the author is assumed first to choose which topics to cover, i.e., which are the elite terms and which are not. Furthermore, it is assumed that frequencies of terms on both the elite and the non-elite set follow a Poisson distribution, with two different means; in other words, for a given term t and for $e \in \{0, 1\}$ (denoting whether we are considering the term to be in the elite or not), there is a random variable $E_{t,e}$ that expresses the distribution of the frequencies of the (elite or non-elite) term t in a document, and $E_{t,e} \sim \text{Poisson}(\lambda_{t,e})$; clearly, we expect $\lambda_{t,1} > \lambda_{t,0}$ (i.e., a single term will appear more frequently if it is elite than if it is not). Plugging this assumption in the general formula derived above determines a monotonic weighting function with an horizontal asymptote that may be interpreted as a form of saturation: the probability of relevance increases with term frequency, but the amount of increase is ultimately close zero when the frequency becomes large. In practice, this is well approximated by

$$w_{t,d}^{\text{BM25}} = \frac{\hat{\mathsf{tf}}_{t,d}}{\hat{\mathsf{tf}}_{t,d} + k_1} \cdot w_t^{\text{idf}}$$

where w_t^{idf} is the inverse document frequency for term t (that determines the asymptotic behavior when the frequency goes to infinity), k_1 is a parameter and $\hat{\mathsf{tf}}_{t,d}$ is a normalized term frequency with respect to the document length, i.e.,

$$\hat{\mathsf{tf}}_{t,d} = \frac{\mathsf{tf}_{t,d}}{(1-b) + b \cdot |d|/\text{avdl}}\ , \tag{1}$$

where $|d|$ is the length of document d, avdl the average length of documents in the collection, and $b \in [0, 1]$ is a tunable parameter. Putting things together, the weight derived from the 2-Poisson elite assumption $w_{t,d}^{\text{BM25}}$ is $U^{\text{BM25}}(\hat{\mathsf{tf}}_{t,d})$ where

$$U^{\text{BM25}}(x) = \frac{x}{x + k_1}.$$

Regions and virtual regions.

In the following derivation we start again from the above-mentioned estimation of

$$w_{t,d}^{\text{BM25}} = \log \frac{p(\mathsf{tf}_{t,d} = x \mid r) \cdot p(\mathsf{tf}_{t,d} = 0 \mid \overline{r})}{p(\mathsf{tf}_{t,d} = x \mid \overline{r}) \cdot p(\mathsf{tf}_{t,d} = 0 \mid r)}.$$

Here, the only features that we need to observe are term frequencies; this is a quite mild assumption and stems from the idea that documents are a single body of text with no structure whatsoever. Some earlier refinements of the probabilistic model, however, already introduced the idea that documents may have some structure. In BM25F [33], for example, a notion of region was introduced: each document is made up of regions, or fields, (e.g., title, body, footnotes etc.) and it is possible to observe term frequencies separately in each region. This extension accounts for the fact that some fields may be more predictive for relevance than others (for example, title will be more predictive than footnotes), and fits well in the eliteness model. The idea is that eliteness of terms is decided beforehand, for every given document, and it is the same across fields; term-frequency, instead, will be again modeled as in standard BM25, although it will be influenced by the length of each field—of course, shorter fields (such as title) are expected to contain more elite terms than longer ones.

As said, the present paper extends this idea by defining what we call *virtual regions*. Ideally, suppose that you have some way (as yet unspecified) to single out some parts of the document that you know will provide high-quality information about relevance; then, you may think of that area as a single virtual region, and apply to that region the evaluation described for BM25F. Those virtual regions are actually obtained by the query itself, through a process that we shall now describe.

An *operator* is a function that associates, to a given query and a given document, another document[2] called a "virtual region". We are given a set of such operators, each endowed with a weight, $\{\langle \Phi_j, w_j \rangle\}_{j \in \mathcal{M}}$: for each query q and document d, we individuate the virtual regions $\Phi_j(d,q)$ (the region of document d matching query q according to operator Φ_j), and for each of them we count the term frequency of each term $t \in q$ in that virtual region; such a frequency is denoted by $\mathsf{tf}_{t,d}^{q,j}$ (the term frequency of term t in the virtual region of document d matching query q according to operator Φ_j); as it is customary in the RSJ model [31], we shall omit the query q, and just write $\mathsf{tf}_{t,d}^{j}$.

To capture this idea, as with BM25F, we proxy the dependence of the eliteness of the occurrences using a set $\{\Theta_j\}_{j \in \mathcal{M}}$ of Bernoulli random variables, which reflect the probability of occurrences in each region generated from a particular operator. De Finetti [8] proved that any set of exchangeable random variables has a representation as a mixture distribution, in general an infinite mixture. Therefore we represent the operators as

$$p(\mathsf{tf}_{t,d}^{j} = x \mid r) = \sum_{e \in \{0,1\}} p(\mathsf{tf}_{t,d}^{j} = x \mid e, r) p(e \mid r) =$$

$$\sum_{q \in Q} \sum_{e,\theta \in \{0,1\}} p(\mathsf{tf}_{t,d}^{j} = x \mid \Theta_j = \theta, e, r) p(\Theta_j = \theta \mid e, r) p(e \mid r)$$

and similarly for $p(\mathsf{tf}_{t,d}^{j} = x \mid \overline{r})$. For $x > 0$, the first of the three factors in the summation is zero if $\theta = 0$, otherwise it is $\lambda_{t,e}^{x} \cdot \mathrm{e}^{-\lambda_{t,e}} / x!$ (for the Poisson-mixture assumption). The second factor for $\theta = 1$ is just β_j, the parameter that governs the j-th Bernoulli Θ_j. The last factor is the probability that the document is elite for the term if it is relevant.

Let us write λ (μ) for $\lambda_{t,1}$ ($\lambda_{t,0}$, respectively) and let p (q) be the probability that a document is elite for the term, given that it is relevant (irrelevant, respectively).

So

$$p(\mathsf{tf}_{t,d}^{j} = x \mid r) = \frac{\lambda^x}{x!} \mathrm{e}^{-\lambda} \beta_j p + \frac{\mu^x}{x!} \mathrm{e}^{-\mu} \beta_j (1-p),$$

and similarly

$$p(\mathsf{tf}_{t,d}^{j} = x \mid \overline{r}) = \frac{\lambda^x}{x!} \mathrm{e}^{-\lambda} \beta_j q + \frac{\mu^x}{x!} \mathrm{e}^{-\mu} \beta_j (1-q).$$

Now following Robertson [30], we can divide both probabilities by $\lambda^x \mathrm{e}^{-\lambda} / x!$ getting

$$\frac{p(\mathsf{tf}_{t,d}^{j} = x \mid r)}{p(\mathsf{tf}_{t,d}^{j} = x \mid \overline{r})} = \frac{\beta_j p + \left(\frac{\mu}{\lambda}\right)^x \mathrm{e}^{\lambda - \mu} (1-p)}{\beta_j q + \left(\frac{\mu}{\lambda}\right)^x \mathrm{e}^{\lambda - \mu} (1-q)}.$$

Observe that, since $\lambda > \mu$, the latter tends to p/q as $x \to \infty$, as in [30].

[2]As explained in the next section, technically the operator produces a set of queries that is then matched against the document to obtain a virtual region, but the difference is immaterial for the moment.

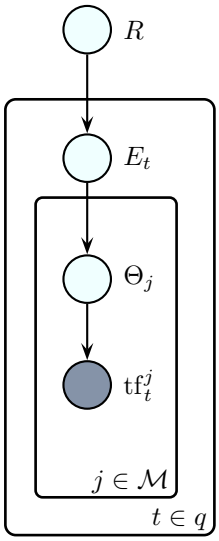

Figure 3: Plate diagram with the extended event space.

The treatment for the case $x = 0$ is slightly more involved, because

$$p(\mathsf{tf}_{t,d}^{j} = 0 \mid r) = \mathrm{e}^{-\lambda} \beta_j p + \mathrm{e}^{-\mu} \beta_j (1-p) + (1-\beta_j)$$

and similarly for the case of irrelevant documents: the last summand depends on the fact that a term can have zero occurrences in a region simply because of Θ_j; so

$$\frac{p(\mathsf{tf}_{t,d}^{j} = 0 \mid \overline{r})}{p(\mathsf{tf}_{t,d}^{j} = 0 \mid r)} = \frac{\mathrm{e}^{-\lambda} \beta_j q + \mathrm{e}^{-\mu} \beta_j (1-q) + (1-\beta_j)}{\mathrm{e}^{-\lambda} \beta_j p + \mathrm{e}^{-\mu} \beta_j (1-p) + (1-\beta_j)}$$

or equivalently

$$\frac{\mathrm{e}^{\mu - \lambda} q + (1-q) + \frac{1}{\beta_j} - 1}{\mathrm{e}^{\mu - \lambda} p + (1-p) + \frac{1}{\beta_j} - 1}.$$

This term behaves like $(1-q)/(1-p)$, under the usual assumption [30] that $\mu - \lambda$ is small, and assuming further that β_j is sufficiently close to 1.

4.1 Implementation of the system

We observe that our underlying retrieval system has a rich query language, the set of whose queries is denoted by Q. Every document d can be matched against the query $q \in Q$ producing a sub-document $M(d,q)$ (i.e., a subsequence of the words the document d is made of).[3] This function is naturally extended to sets of queries, by letting $M(d,A)$ be the union of all sub-documents $M(d,q)$ for $q \in A$.

Conversely, a *raw query* is just a sequence of terms $r = \langle t_1, \ldots, t_u \rangle$: this is what we suppose that the user inputs to the system; the set of all raw queries is denoted by R.

An *operator* is a function $\Phi : R \to 2^Q$ mapping a raw query to a set of queries. Here are some simple examples of operators that we will be using in our experiments:

[3]The exact semantics of the match depends on the query language and on the retrieval system used and will be not described further, but see [9, 13] for two examples.

- $\Phi_{\text{bag-of-words}}$: maps a raw query $\langle t_1, \ldots, t_u \rangle$ to the set of queries whose element are the single terms (i.e., to $\{t_1, \ldots, t_u\}$);
- $\Phi_{p\text{-grams}}$: maps a raw query $\langle t_1, \ldots, t_u \rangle$ to the set of all p-grams of p consecutive terms in the query (i.e., to $\{\text{“}t_1 \cdots t_p\text{”}, \text{“}t_2 \cdots t_{p+1}\text{”}, \ldots, \text{“}t_{u-p+1} \cdots t_u\text{”}\}$);
- $\Phi_{p\text{-AND}}$: maps a raw query $\langle t_1, \ldots, t_u \rangle$ to the set of all conjunctions of p arbitrary terms in the query (e.g., for $p = 2$, to $\{t_i \text{ AND } t_j \mid i \neq j\}$);
- Φ_{phrasal}: maps a raw query $\langle t_1, \ldots, t_u \rangle$ to the single phrasal query $\text{“}t_1 \cdots t_u\text{”}$;
- Φ_{segments}: maps a raw query to the set of consecutive terms that make up a concept (see Section 5 for a complete description).[4]

Let now $\{\langle \Phi_j, w_j \rangle\}_{j \in \mathcal{M}}$ be a set of operators and weights; for a fixed raw query $r = \langle t_1, \ldots, t_u \rangle$, let $\mathsf{tf}^j_{t,d}$ be the number of occurrences of term t in $M(d, \Phi_j(r))$. The average term frequency of term t in document d is then defined to be[5]

$$\hat{\mathsf{tf}}_{t,d} = \sum_{j \in \mathcal{M}} \frac{w_j \cdot \mathsf{tf}^j_{t,d}}{(1 - b_j) + b_j \cdot |d|/\text{avdl}}.$$

The score of document d for query q is then computed as

$$\sum_{t \in q} \frac{\hat{\mathsf{tf}}_{t,d}}{\hat{\mathsf{tf}}_{t,d} + k_1} \cdot w_t^{\text{idf}},$$

where the latter is the standard term idf.

BM25(F) as a special case.

Note that using a single bag-of-words operator reduces our scoring formula to the usual BM25 score. Conversely, suppose your collection has G fields, and let $\Phi_1, \ldots, \Phi_G$ are operators that work like a standard bag-of-words, but where Φ_i tries to find matches only in field i. So, for example, $M(d, \Phi_{\text{title}}(t))$ would return the sub-document of the title made only by the occurrences of term t. Then, an application of the above formula would reduce to the standard BM25F score.

4.2 Remarks and variants

BM25, following the original probabilistic relevance framework, adopts a disjunctive semantics, that is, there is no need for a document to contain *every* query term in order to receive a non-zero score. The key point behind this assumption is that we need an external mechanism to decide which eliteness models we want to take into account for each query, and this will simplify the number of estimations and scores we need to compute. After that, it is important to decide what is an appropriate shape of the functional estimating relevance probability as a function of term scores: in Section 3 we showed empirically that conjunctive and proximity operator produce the same shape as in the 2-Poisson eliteness model.

An issue raised by our model, and that we must take into account, is the fact that the very same occurrence of a term within a document will be counted more than once, because virtual regions (differently from BM25F fields) may overlap. For instance, if we want to score separately matches of stemmed query terms and matches of unstemmed query terms we would be double-scoring some of the occurrences. This remark calls for discounting signals coming from the same source; one way to obtain this result would be to establish some dependence between the operators Φ_j: in the example above, we might correct the estimation using a $p(\Theta^{\text{st}}_j \mid \Theta^{\text{ex}}_j, r)$ correction factor (here, and in the following, the superscripts st and ex stand for “stemmed” and “exact”, respectively).

In practice, however, we can avoid this estimation by recalibrating the different weights. We empirically know that both exact and stemmed matches should contribute to the score on which the saturation function is applied, and we want to aggregate those contributions together, using the proper weights. We can then correct the double counting and substitute accordingly in equation (1) as

$$\hat{\mathsf{tf}}_{t,d} = \frac{w^{\text{ex}} \cdot \mathsf{tf}^{\text{ex}}_{t,d} \cdot w^{\text{ex}}_{\text{idf}}}{1 - b_0 + b_0 \cdot |d^{\text{ex}}|/\text{avdl}^{\text{ex}}} + \frac{w^{\text{st}} \cdot \mathsf{tf}^{\text{st}}_{t,d} \cdot w^{\text{st}}_{\text{idf}}}{1 - b_1 + b_1 \cdot |d^{\text{st}}|/\text{avdl}^{\text{st}}}$$

Some observations about possible variants of our model are worth being remarked here:

- We could have used a different saturation function at the eliteness level; in fact, as with BM25, we could in principle learn the real function shape using an appropriately large dataset (but we would run the risk of over-fitting, though). The one we adopted $\hat{\mathsf{tf}}/(\hat{\mathsf{tf}} + k)$ is appealing: it resembles a logistic function passing through the origin and when the class conditional distributions $p(x|r)$, $p(x|\overline{r})$ belong to the same exponential family, the log-odds ratio of the class posteriors $\log\left(\frac{p(r|x)}{p(\overline{r}|x)}\right)$ will belong to a logistic family [4] (see also [25] for a connection of the log-logistic model and the term frequency normalization of BM25).
- We could have tackled the problem of incorporating positional information within the probabilistic framework by coming up with a higher-order model adding the positions of terms as variables; as noticed, though, this approach would lead either to an *ad-hoc* kernel-like method or to an exponential number of parameters to estimate. Given the scarcity of publicly available training data, and for the sake of domain transparency, we believe that the proposed framework provides an acceptable solution and a good trade-off between learning requirements, complexity and performance.
- The operators that form the basis upon which our system builds can be introduced in many ways: a set of operators can be fixed and applied silently to all queries introduced by the user, as a form of multi-level enrichment (this is what we are assuming in the rest of the paper), or it can be defined externally on a

[4]Both the p-gram, the phrasal and the segment operators can be endowed with an enlargement factor that allows for some extra word to sneak in—also this point will be fully explained in Section 5.

[5]The length-normalization factor here might actually be different for each virtual region, but this solution turns out to be extremely expensive to implement, because the average region length is unknown unless the whole collection is examined. A good approximation can be obtained by using the standard document length as a measure of “verbosity” of all the virtual regions it contains.

per-query basis, or on a per-query-type basis; we may instead think that it is the user herself to introduce operators in the query, which may be sensible and may find applications in some contexts where users are expected to adopt a richer query language. Finally, it is certainly possible to mix the two approaches, having some operators introduced directly by the user and others added automatically by the system.

5. QUERY CONCEPT SEGMENTATION

Albeit today's search engines offer a limited number of operators (phrasal, proximity etc.) most users tend to stick at introducing plain queries, typically a sequence of few words (about 3.08 on average, according to recent studies [36]). Nonetheless, as observed previously [37], many queries are actually made up by some basic conceptual units, each of them being formed possibly by many words. For example, the query `indian summer victor herbert` is clearly formed by two conceptual elements ("indian summer" and "victor herbert", the former referring to a meteorological phenomenon, the second to a person); breaking such conceptual units apart, or inverting the order of the words that label them, would produce information loss—of course, many documents will contain both the words "indian" and "summer" without referring to "indian summer"; also "summer indian" will also probably appear in some document, for example in reference to summer indian food, without any relevance to the concept sought. In some cases, it may be wise to allow for one spurious word to be inserted within a segment (so that, for example, `san jose airport` can also match the sequence "San Jose international airport", or `george bush` match "George W. Bush").

Among our goals, we would like the model to accommodate for query term-dependence, or concept-detection, at the query level. For example, if there is evidence that the query `san jose aiport` consists of two concepts, one referring to a city and the other one denoting a place or an action, we would like the model to be able to weight differently documents that only match one concept from those that match both, taking also into account the distance between the terms making up the concepts. To this end, one of our operators will employ query segmentation, an emerging NLP task that aims at identifying sub-sequences of strings that refer to a single unit or concept [7, 37], as described above. There have been limited attempts to integrate segments into ranking [5, 6, 20] in the context of the language modeling framework [39], but to the best of our knowledge this is the first time that a similar attempt is being applied to BM25-like techniques.

In the experimental section, we use both a segmentation operator based on generative language models and Wikipedia (as described in [37]) and a simple operator extracting p-grams of consecutive terms: the latter is of course less precise (because some p-grams do not correspond to concepts), and is given a lower weight — it serves the purpose of identifying possible word-sequences that for some reason the segmentation algorithm failed to guess. Moreover, for both types of operators we allowed for a variety of amount of spurious words appearing in the segment: this is obtained by introducing different *enlargement factors* μ (a multiplicative factor determining the maximum allowed ratio between the length of the span found and the length of the segment); an enlargement factor $\mu = 1$ corresponds to accepting only exact matches, whereas for example allowing for an enlargement $\mu = 2$ corresponds to accepting at most one extra word for every single word in the segments (e.g., if the segment is made up of two words, it is still acceptable if we find them two words apart). Different enlargement factors are used in the experiments, of course assigning larger weights to smaller enlargements.

Collection	Size	Documents	Topics
TERA04	436G	25M	TREC 701-750
TERA05	436G	25M	TREC 751-800
TERA06	436G	25M	TREC 801-850
WEB	100G	10M	1000
WEB-Phrasal	100G	10M	400

Table 1: Data collections

6. EXPERIMENTAL RESULTS

We tested the usefulness and accuracy of the operators described in Section 4.1 and 5 in a series of experiments. We posit that query segments will only affect to a limited number of queries in the data-set; however the working question is whether combining them has some positive retrieval effect or not. Retrieval performance is optimized iteratively using MAP as a target metric. We sweep one parameter at a time over the allowed parameter range, holding every other parameter fixed, in a similar fashion to the method of Metzler and Croft [26]. Each operator introduces two parameters: a weight w_j controlling the relative importance of the operator, and a factor b_j determining the impact of term-frequency normalization with respect to the corresponding operator. Different *enlargement factors* μ (for segments, p-grams and phrasal operators) are reported in separate rows of the table, as well as different values for the number p of consecutive terms that are considered while building p-grams.

We report on MAP and P@10 and check for statistical significance using a one-sided t-test with $p < 0.05$. The operators are trained on two different Web collections: GOV2, TREC's Terabyte track collection [14, 15, 12], and a subsample of the Web from 2011. The collections are described in Table 1. Training and testing is performed on different topic sets; for TREC topics, we train and test on different years (train on TERA04, test on TERA05/06; train on TERA05, test on TERA04), and for the Web collection we perform 10-fold cross validation across the whole topic set. For the TREC collection, we used the topic title as raw query.

Table 2 presents the results of applying one single operator, combined with BM25 (that is, with the bag-of-words operator). Experiments show that all the operators are able to improve the performance on their own to a reasonable extent. The impact of single operators, specially the segments, is limited. However, it is worth noting that the methods using a single operator that restrict the semantics of the matching (like AND or segments) perform comparably to the best values reported at TREC for early precision (P@20) [14, 15, 12]. Table 3 presents the results of combining segment and p-gram operators with BM25 and BM25F. The different operators have been chosen using various enlargement factors; we further enriched the segment combination with a 2-gram

		TERA04			TERA05			TERA06		
		MAP	P@10	P@20	MAP	P@10	P@20	MAP	P@10	P@20
BM25		0.2648	0.5327	0.5071	0.3228	0.6140	0.5600	0.2928	0.5380	0.5140
BM25F		0.2697	0.5510	0.5143	0.3284	0.6200	0.5570	0.2935	0.5460	0.5160
p-AND	$p = 2$	0.2679	0.5429	**0.5286**	**0.3396***	0.6260	**0.5920**	0.3069	0.5900	0.5300
phrasal	$\mu = 3$	0.2673	0.5306	0.4939	0.3369*	0.6060	0.5790	0.3082*	0.5720	0.5290
segment	$\mu = 1$	0.2685*	0.5143	0.4970	0.3274	0.6080	0.5580	0.3180*	0.5860	0.5350
segment	$\mu = 1.5$	0.2695*	0.5347	0.5010	0.3272	0.6140	0.5620	0.3122*	**0.5940**	0.5460
segment	$\mu = 3$	0.2690*	0.5143	0.5204	0.3295*	**0.6300**	0.5840	**0.3277***	**0.5940**	0.5460
p-grams	$p = 2,\mu = 1$	**0.2786***	**0.5530**	0.5120	0.3294	0.5860	0.5470	0.3137*	0.5860	0.5470
p-grams	$p = 3,\mu = 1$	0.2670	0.5060	0.4950	0.3272	0.5980	0.5560	0.3198*	0.6160	**0.5600**

Table 2: Performance of single operators (* = statistical significance at $p < 0.05$ using a one-sided t-test with respect to BM25, MAP only).

operator. The purpose of this experiment is to determine whether the sigmoid-like term-frequency normalizing function is able to accommodate for different features which stem from the same source of evidence (matches of the query term in the document). Results are significantly better than the baseline and outperform state-of-the-art ranking functions that just use matching of query terms (note we are not adding query-independent evidence, like link information, click-through data, etc.). For instance, the best MAP value for TREC 2004 at TERA04 [14] was 0.2844, the sequential dependence model of the Markov random field for IR peaks at MAP 0.2832 [26] and the two-stage segmentation model of Bendersky *et al.* had an average MAP of 0.2711 over the 150 topics [5].

In order to explore further the usefulness of query segmentation and p-grams for *difficult* queries, we selected a subsample of 1000 queries taken from Yahoo! Search; the data corpus was a 100GB sub-sample of the Web. The queries had been evaluated by a trained editorial team, with about 130 judged documents per query on average: relevance was assigned on a 4-level scale, from *Bad* to *Excellent*; given that we had graded relevance available, we report on NDCG (gain values at the relevance level, from 0 to 4) and MAP. The average query length was 3.14. We report the performance of BM25 and BM25F as baselines. In addition, we selected a sub-sample of 400 queries where the user herself had employed query segments (i.e., phrasal queries), and removed the segmentation information. This experiment was aimed at establishing if a more elaborate query interpretation mechanisms is able to be of help in these cases.

Table 4 shows that segmentation and p-grams, when mixed with the operator combination are able to improve the performance of a large number of Web queries. When looking at the differences between the regular and phrasal queries, we observe that gains, even if consistent over the two different sets, are slightly higher in the second group ($\approx$ 12.3% vs. $\approx$ 7.1% in MAP for p-grams vs. BM25F). This fact indicates that the method is helpful for queries that contain *difficult* concepts, as they have been expressed by users manually, whereas maintaining the performance of queries in which identifying concepts is not so critical and standard bag-of-word approaches perform as well.

Anecdotal study.

It is useful to look into the reasons behind the increased retrieval performance we observed in our experiments; with this goal, we considered separately the mean average precision on each of the TREC topics 701-850 and examined the ten queries that produced the largest difference in precision between our system and the BM25 baseline. Table 5 shows the queries that obtained the greatest benefit from the use of segmentation and/or p-gram operators: in the rightmost column of the table, we identified the segments (or the p-grams) that most contributed to the increased precision, allowing to retrieve relevant documents that BM25 missed (because they were ranked too low).

7. CONCLUSIONS AND FUTURE WORK

In this paper we proposed a way to extend the probabilistic relevance framework with a notion of *virtual region* based on the use of operators applied to the query. Our experiments (both on standard collections, such as TREC, and on other Web-like repertoires) show that the use of virtual regions is especially beneficial for hard queries where positional information is actually precious. The method has room for improvements and further study should be undertaken to understand which operators are more useful and under which circumstances. Even if we explored a reasonable number of combinations, we have not made any systematic attempt to develop a method to select the individual best operators; we just limited ourselves to handpick a few—it was out of the scope of this paper to analyze automated operator selection. In contrast, a machine learning approach [3] would derive features for as many operators as possible and try to combine them optimizing a loss function of MAP or NDCG [38]. One might even envisage the adoption of a query-classification tool to decide which operators should be used, based on the presumed nature of the query. In any case, our experiments show the practical usefulness of the non-linear operator score combination for retrieval [33]. As a final remark, it is interesting to observe that most of the studied operators (actually, all of them except for segments) do not employ *any* source of external information, and still produce a significant performance improvement.

8. REFERENCES

[1] E. Agichtein, E. Brill, S. Dumais, and R. Ragno. Learning user interaction models for predicting web search result preferences. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, volume 1, pages 3–10. ACM, 2006.

[2] R. Baeza-Yates. Query intent prediction and recommendation. In *Proceedings of the fourth ACM*

	TERA04			TERA05			TERA06		
	MAP	P@10	P@20	MAP	P@10	P@20	MAP	P@10	P@20
BM25	0.2648	0.5327	0.5071	0.3228	0.6140	0.5600	0.2928	0.5380	0.5140
BM25F	0.2697	0.5510	0.5143	0.3284	0.6200	0.5570	0.2935	0.5460	0.5160
BM25 + p-grams	0.2898*	0.5939	0.5531	0.3368*	0.6260	0.5800	0.3361*	0.6000	0.5048
BM25 + segment	0.2866*	0.5653	0.5306	0.3285	0.5980	0.5710	0.3188	0.5720	0.5100
BM25F + p-grams	**0.2908***	**0.5959**	**0.5541**	**0.3398***	**0.6280**	**0.5830**	**0.3319***	**0.5940**	**0.5350**
BM25F + segment	0.2869*	0.5653	0.5306	0.3282*	0.5920	0.5510	0.3315*	**0.5940**	**0.5350**

Table 3: Performance of a combination of operators (* = statistical significance at $p < 0.05$ using a one-sided t-test with respect to BM25, MAP only). Configuration for p-gram is $p = 2, \mu = \{1, 2, 3\}$, and for segment is $\mu = \{1, 2, 3\}$.

	WEB			WEB-phrasal		
	MAP	NDCG	P@10	MAP	NDCG	P@10
BM25	0.1553	0.2728	0.3892	0.1350	0.2345	0.3133
BM25F	0.1722*	0.3008	0.4285	0.1575*	0.2842	0.3747
BM25 + p-grams	0.1822*	0.3126	0.4390	0.1750*	0.3078	0.4010
BM25 + segment	0.1810*	0.3126	0.4344	0.1725*	0.3029	0.4040
BM25F + p-grams	**0.1854***	**0.3216**	**0.4400**	**0.1769***	**0.3123**	**0.4101**
BM25F + segment	0.1842*	0.3190	0.4392	0.1749*	0.3165	0.4005

Table 4: Performance of a combination of operators on the WEB collection (* = statistical significance at $p < 0.05$ using a one-sided t-test with respect to BM25, MAP only). Configuration for p-gram is $p = 2, \mu = \{1, 2, 3\}$, and for segment is $\mu = \{1, 2, 3\}$.

conference on Recommender systems, pages 5–6. ACM, 2010.

[3] B. Bai, J. Weston, D. Grangier, R. Collobert, K. Sadamasa, Y. Qi, O. Chapelle, and K. Weinberger. Learning to rank with (a lot of) word features. *Information Retrieval*, 13:291–314, 2010. 10.1007/s10791-009-9117-9.

[4] A. Banerjee. An analysis of logistic models: exponential family connections and online performance. In *SIAM International Conference on Data Mining*, 2007.

[5] M. Bendersky, B. Croft, and D. a. Smith. Two-stage query segmentation for information retrieval. *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval - SIGIR '09*, page 810, 2009.

[6] M. Bendersky, D. Metzler, and W. Croft. Learning concept importance using a weighted dependence model. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 31–40. ACM, 2010.

[7] S. Bergsma and Q. Wang. Learning noun phrase query segmentation. In *Proc. of EMNLP-CoNLL*, number June, pages 819–826, 2007.

[8] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.

[9] P. Boldi and S. Vigna. MG4J at TREC 2005. In E. M. Voorhees and L. P. Buckland, editors, *The Fourteenth Text REtrieval Conference (TREC 2005) Proceedings*, number SP 500-266 in Special Publications. NIST, 2005. `http://mg4j.dsi.unimi.it/`.

[10] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, Apr. 1998.

[11] S. Büttcher, C. L. A. Clarke, and B. Lushman. Term proximity scoring for ad-hoc retrieval on very large text collections. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 621–622, New York, NY, USA, 2006. ACM.

[12] S. Büttcher, C. L. A. Clarke, and I. Soboroff. The TREC 2006 terabyte track. In *TREC*, 2006.

[13] C. L. A. Clarke, G. V. Cormack, and F. J. Burkowski. An algebra for structured text search and a framework for its implementation. *The Computer Journal*, 38:43–56, 1995.

[14] C. L. A. Clarke, N. Craswell, and I. Soboroff. Overview of the TREC 2004 terabyte track. In *TREC*, 2004.

[15] C. L. A. Clarke, F. Scholer, and I. Soboroff. The TREC 2005 terabyte track. In *TREC*, 2005.

[16] S. Cooper. Some Inconsistencies and Misnomers Retrieval in Probabilistic Information of Library taken Effective. *Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 57–61, 1991.

[17] N. Craswell, S. Robertson, H. Zaragoza, and M. Taylor. Relevance weighting for query independent evidence. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, volume 1, pages 416–423. ACM, 2005.

[18] P. Domingos, S. Kok, H. Poon, M. Richardson, and P. Singla. Unifying logical and statistical AI. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence*, pages 2–7, 2006.

[19] M. Hagen, M. Potthast, B. Stein, and C. Braeutigam. The power of naive query segmentation. In *Proceeding of the 33rd international ACM SIGIR conference on*

Topic no.	BM25 MAP	Operators MAP	Query	Which segment / p-gram helped?
810	0.2759	0.5243	Timeshare resales	"Timeshare resales"
750	0.0903	0.3251	John Edwards womens' issues	"John Edwards" and "womens' issue"
806	0.1279	0.3446	Doctors Without Borders	"Doctors Without Borders"
834	0.2819	0.4859	Global positioning system earthquakes	"Global positioning"
745	0.1801	0.3828	Doomsday cults	"Doomsday cult"
835	0.1306	0.3222	Big Dig pork	"Big Dig"
732	0.1438	0.3309	U.S. cheese production	"U.S." and "cheese production"
723	0.1187	0.3030	Executive privilege	"Executive privilege"
843	0.2420	0.4228	Pol Pot	"Pol Pot"
785	0.2717	0.4323	Ivory billed woodpecker	"Ivory billed"

Table 5: The ten queries that produced the largest gap between standard BM25 and our operator-based scoring. The methods in this table use a combination of BM25, segments and 2-grams.

Research and development in information retrieval, pages 797–798. ACM, 2010.

[20] Y. Li, B.-j. P. Hsu, C. Zhai, and K. Wang. Unsupervised Query Segmentation Using Clickthrough for Information Retrieval. *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 285–294, 2011.

[21] T.-Y. Liu. Learning to Rank for Information Retrieval. *Information Retrieval*, 3(3):225–331, 2009.

[22] Y. Lu, F. Peng, G. Mishne, X. Wei, and B. Dumoulin. Improving web search relevance with semantic features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing Volume 2 - EMNLP '09*, number August, page 648, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

[23] Y. Lv and C. Zhai. Positional language models for information retrieval. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 299–306. ACM, 2009.

[24] Y. Lv and C. Zhai. Positional relevance model for pseudo-relevance feedback. In *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 579–586. ACM, 2010.

[25] Y. Lv and C. Zhai. A log-logistic model-based interpretation of TF normalization of BM25. In *Proceedings of the 34th European Conference on Information Retrieval*, ECIR'12, 2012.

[26] D. Metzler and B. Croft. A Markov random field model for term dependencies. *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '05*, page 472, 2005.

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

[28] L. Ramshaw. Text chunking using transformation-based learning. *of the Third ACL Workshop on Very*, pages 82–94, 1995.

[29] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62(1):107–136, 2006.

[30] S. Robertson. The probability ranking principle in IR. *Journal of documentation*, 1977.

[31] S. Robertson. On event spaces and probabilistic models in information retrieval. *Information Retrieval*, 8(2):319–329, 2005.

[32] S. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. *Proceedings of the SIGIR conference on Research and development in information retrieval*, (1), 1994.

[33] S. Robertson, H. Zaragoza, and M. Taylor. Simple BM25 extension to multiple weighted fields. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, CIKM '04, pages 42–49, New York, NY, USA, 2004. ACM.

[34] S. E. Robertson and H. Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009.

[35] K. Svore, P. Kanani, and N. Khan. How good is a span of terms?: exploiting proximity to improve web retrieval. In *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 154–161. ACM, 2010.

[36] M. Taghavi, A. Patel, N. Schmidt, C. Wills, and Y. Tew. An analysis of web proxy logs with query distribution pattern approach for search engines. *Computer Standards and Interfaces*, 34(1):162 – 170, 2012.

[37] B. Tan and F. Peng. Unsupervised query segmentation using generative language models and wikipedia. In *Proceeding of the 17th international conference on World Wide Web*, pages 347–356. ACM, 2008.

[38] M. Taylor, J. Guiver, S. Robertson, and T. Minka. Softrank: optimizing non-smooth rank metrics. In *Proceedings of the international conference on Web search and web data mining*, WSDM '08, pages 77–86, New York, NY, USA, 2008. ACM.

[39] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems*, 22(2):179–214, Apr. 2004.