Combining concepts and language models for information access

Meij, E.J.

Citation for published version (APA):
Meij, E. J. (2010). Combining concepts and language models for information access.

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (http://dare.uva.nl)
Experimental Methodology

In the previous chapter we have looked at the assumptions, models, and related work underpinning this thesis. In this chapter we introduce the experimental methodology generally employed in IR and also adopted in this thesis. We start by discussing the notion of relevance, detailing standard forms of evaluation, and significance testing. We then describe the data sets that will be used in later chapters. We conclude with a section discussing retrieval model parameters.

3.1 Relevance

Central to the evaluation of IR systems is the notion of relevance. Relevance of a piece of information (be it a web page, document, passage, or anything else) is measured against an information need of some user. Contextual factors such as presentation or document style aside [133], determining a topical definition of an information need is subject to various user-based parameters [159]. For example, different users may have different backgrounds, their understanding of the topic might change as they browse through a result list, or they may aim to solve different tasks. Objectively determining relevance of a piece of information to an information need is difficult to operationalize. Cool et al. [78], for example, studied the real life tasks of writing an essay and found that characteristics other than topical relevance affect a person’s evaluation of a document’s usefulness. This complexity of relevance as an evaluation criterion has been recognized already by Saracevic [279] and is still pertinent today.

Cooper [79] posits that any valid measure of IR system performance must be derived from the goal of such a system. Since the goal is to satisfy the information need of a user, a measure of utility to the user is required. Cooper concludes that user satisfaction with the results generated by a system is the optimal measure of performance. These intuitions provide the basis for the user-based approach to IR system evaluation. According to this view, systems should be evaluated on how well they provide the information needed by a user. And, in turn, the best judge of this performance is the person who is going to use the information. De-
spite criticisms [289], researchers committed to a user-centered model of system evaluation.

The Cranfield experiments sidestepped any issues pertaining to relevance [74, 75, 260]. In Cranfield I, queries were generated from documents and the goal was to retrieve the document each query was generated from. As such, there was only a single relevant document to be retrieved for each query. In Cranfield II, queries were generated in the same way, but each document was now manually judged for relevance. In a recent study, Kelly et al. [163] report on the results of a user study. They find that there exists linear relationships between the users’ perception of system performance and the position of relevant documents in a search results list as well as the total number of retrieved relevant documents; the number of relevant documents retrieved was a stronger predictor of the users’ evaluation ratings. In the next section we introduce the common methodology associated with the evaluation of IR systems.

3.2 Evaluation

The evaluation of IR systems has a long tradition, dating back from before the Cranfield experiments [75, 164, 260]. It is an important part of the experimental methodology to determine how well IR systems satisfy users’ information needs and whether some system does this better than another [309, 325]. There are several publications addressing various aspects of evaluation. Voorhees and Harman [332] detail the history of TREC and the evaluation methods used there. Harman [120] gives an overview of the state of IR evaluation in 1992. More recently, Robertson [260] provided his personal view on the history of evaluation for IR. Sanderson [277] gives an overview of current methods and practices. Tague-Sutcliffe [309] defines six elements that comprise the IR process:

1. a document set to be searched (the “collection”),
2. a user need,
3. a query (usually called “topic”),
4. a search strategy,
5. a retrieved list of documents, and
6. relevance judgments (typically referred to as “qrels”).

Typically when doing IR evaluation, the retrieval system is given a verbalization of the information need (as a query, ranging from a few keywords to a full narrative) which it uses as input to its retrieval algorithm (the “search strategy”, cf. Section 2.1). The output of this algorithm is a ranked list of documents that may
then be inspected by the user with the information need. It is common to refer to the combination of the document collection, topics, and accompanying judgments as “test collection.”

Ideally, we would like to verify the effectiveness of every system on real life users. However, as already indicated in the previous section, relevance is not a deterministic notion and varies per user, task, setting, etc. This, as well as the prohibitive costs of such evaluations, have resulted in an established tradition of sampling and pooling methods [121, 362]. Evaluation campaigns such as FIRE, TREC, CLEF, NTCIR, and INEX provide systematic evaluations on sets of topics and documents, which are subsequently used to rank IR systems according to their performance. In order to make the evaluation tractable, pooling of the results of the participating systems is applied. Here, the top-ranked documents up to a certain rank are taken from each participating system and judged for relevance. Although not all documents in the collection are judged for relevance using this approach, it was found that systems could still be reliably evaluated using this approach [332]. Moreover, even systems not contributing to the pools could still be fairly assessed [362]. Whether these findings still hold for every retrieval metric on very large document collections is a topic of ongoing research [49, 52]. In the mean time, various alternatives to pooling are investigated [61, 62], as detailed below. A distinct benefit of such system-based evaluations is the reusability of test collections, since future systems can be reliably evaluated and compared using the same assessments [260, 277, 332].

It is common to not evaluate the ranked list itself, but merely the documents that appear in it. Recent work, however, recognizes that the first thing that a user sees and interacts with is the list of retrieved documents [22]. Bailey et al. define a novel evaluation method focusing on this initial interaction and find that it provides a natural complement to traditional, system-based evaluation methods.

With the recent advent of relatively cheap crowdsourcing possibilities such as Amazon’s mechanical turk service, a renewed interest in obtaining relatively cheap, manual relevance assessments for various systems has emerged [5, 7]. Whether such evaluations live up to their premise of cheap, consistent relevance assessments on a substantial scale is as of yet unclear and in the remainder of this thesis we use more traditional, established TREC-style evaluations.

In the following sections, we look at typical IR effectiveness metrics used in this thesis, as well as statistical testing on these measures.

### 3.2.1 Evaluation Measures

Different search tasks exist, each with a different user model. In all of the cases presented in this thesis, a user wants to find information on a topic (topic-finding or ad hoc retrieval). Other cases include users having a specific web page or document in mind (named-page finding), users looking for an answer to a spe-
3. Experimental Methodology

cific question (question answering), users looking for relevant experts or entities (expert/entity finding), or users having a standing information need, where new documents entering in the collection are to be routed to the users with an interest in the topic of the document (adaptive filtering). Each of these search tasks calls for evaluation measures that fit the task. For example, in the case of named-page finding, there is typically only one relevant document (the one that the user has in mind). A proper evaluation measure for this task should reward systems that place that document at the top of the ranking and penalize systems that do not.

Researchers have been considering how to evaluate results originating from a retrieval system for a number of decades now and the choice of measures and their analysis remains an active theme of research. Kent et al. [164] were the first to introduce the notion of recall and precision. These intuitive measures consider the documents retrieved in response to a user’s query as a set and indicate the fraction of retrieved documents that are relevant (precision) or the fraction of relevant documents retrieved (recall) [202]. These measures are best explained through the use of a contingency (or confusion) table, cf. Table 3.1. In this table, the documents are split by whether they are retrieved by a system and whether they are relevant. Precision, then, is defined as:

\[
\text{Precision} = \frac{tp}{tp + fp}, \tag{3.1}
\]

whereas recall is defined as:

\[
\text{Recall} = \frac{tp}{tp + fn}. \tag{3.2}
\]

Although precision and recall are set-based measures, they are commonly applied to ranked lists by truncating the lists at a certain rank. A common visualization of these measures is to plot precision values at different levels of recall. The resulting graph is called a precision-recall graph; an example may be found in Figure 5.3 (see page 102).

Given that precision is the ratio of retrieved relevant documents to all documents retrieved at a given rank, the average precision (AP) is defined as the average of precisions at the ranks of relevant documents. More formally, for a set of relevant documents, \( R \):

\[
\text{AP} = \frac{1}{|R|} \sum_{d \in R} \text{prec@rank}(d), \tag{3.3}
\]
where $|R|$ equals the size of the set of known relevant documents for this query. Buckley and Voorhees [49] show that AP is stable; that is, it is able to reliably identify a difference between two systems when one exists. In later chapters, our main evaluation measure is AP averaged over a number of queries, called mean average precision (MAP). These and other measures are obtained using the trec_eval\(^1\) program.

In later chapters we use the following abbreviations for the evaluation measures:

**PX** – Precision at rank \(X\). In the case of P1 this indicates the proportion of queries for which a relevant occurred at rank 1.

**R-prec** – Precision at rank \(|R|\). If this value equals 1, all relevant documents are placed at the top of the ranking.

**MAP** – Mean average precision.

**SRX** – Success rate at rank \(X\); a binary measure that indicates whether at least one correct document has been returned in the top-\(X\) (when there is no rank indicated we assume \(X=5\)). When averaged over a number of queries it indicates the proportion of queries for which a relevant document occurred in the top-\(X\).

**MRR** – The mean of the reciprocal of the rank of the first relevant document.

**RelRet** – The number of relevant documents retrieved (measured at rank 1000, unless indicated otherwise). When this value is expressed as a fraction of the total number of relevant documents, it is called “recall”, cf. Eq. 3.2.

Of these, MRR, PX, and SRX correspond directly to common user experience since they measure the presence and/or amount of relevant documents at the top of the document ranking [262, 286]. Other users, however, may be more interested in retrieving as many relevant documents as possible and, for them, RelRet might be more appropriate. As indicated above, MAP has both a precision and a recall aspect. We will therefore use this measure as our main evaluation metric.

As indicated above, for relatively small document collections it is feasible to collect relevance assessments on all the documents given a query. For larger collections, it is assumed that the top-ranked documents collected from a variety of systems form a reliable basis for evaluation. This in turn enables the comparisons of systems on the basis of recall, which requires the count of all relevant documents for a query. As document collections grow, however, these assumptions may no longer hold [49, 52]. Therefore, several new measures (typically based on a form of sampling or bootstrapping) are being developed for such collections [1, 14, 62, 71]. For the largest document collection that we employ later

\(^1\)See http://trec.nist.gov.
in the thesis (ClueWeb09; introduced in Section 3.3.4), we report these measures instead of the traditional ones. Specifically, for ClueWeb09, Category B we report statMAP and statP10 [14], whereas for ClueWeb09, Category A we also report expected MAP (eMAP), expected R-precision (eR-prec), and expected precision at rank 10 (eP10) [61, 62]. Systems participating in TREC tracks that use ClueWeb09 were pooled up until a relatively shallow depth and these measures are intended to yield the same ranking as traditional measures would have if the runs had been fully judged.

TREC Web 2009 (a test collection that makes use of the ClueWeb09 document collection—see below) featured a novel sub-track, aiming to improve diversity in the result list. The diversity task is similar to the ad hoc retrieval task, but differs in its judging process and evaluation measures. The goal of this task is to return documents that together provide complete coverage for a query, while avoiding excessive redundancy in the result list; the probability of relevance of a document is conditioned on the documents that appear before it in the result list. Each topic is therefore structured as a representative set of subtopics (and unknown to the system). Each subtopic, in turn, is related to a different user need and documents are judged with respect to the subtopics. The evaluation measures associated with diversity that we report upon in the thesis are: $\alpha$-nDCG [71] and intent aware precision@10 (IA-P@10) [1]. The former is based on normalized discounted cumulative gain [146] and rewards novelty and diversity in the retrieved documents. The parameter $\alpha$ indicates the probability that a user is still interested in a document, given that subtopic of the current document has already been covered by the preceding documents. We use the default setting of $\alpha = 0.5$. The second measure is similar to precision@10, but incorporates information from a taxonomy (the ODP taxonomy in particular) to determine diversity.

### 3.2.2 Statistical Significance Testing

As indicated earlier in this chapter, relevance assessments are not deterministic and there is inherent noise in an evaluation. Early work on a small document collection indicated that a large variance in relevance assessments does not have a significant influence on average recall and precision [186]. As test collections grew, however, questions were asked with respect to the validity of this conclusion on larger and more variable test collections [141].

So, given two systems that produce a ranking of documents for a topic, how can we determine which one is better than the other? Our method should be robust and promote the system that is truly better, rather than promoting the one that performed better by chance. Statistical significance testing plays an important role in making this assertion. A significance tests consists of the following three ingredients [287]:

1. **Test statistic**: A measure that is calculated from the data and is used to assess the null hypothesis.
2. **P-value**: The probability of observing the test statistic or something more extreme if the null hypothesis is true.
3. **Critical value**: A threshold value used to determine whether the null hypothesis should be rejected.
3.2. Evaluation

1. A test statistic or criterion by which to judge the two systems. Typically, the mean of a retrieval metric introduced in Section 3.2.1 is used.

2. A distribution of the test statistic given the null hypothesis. The typical null hypothesis (and the one we use in this thesis) is that there is no difference between the systems.

3. A significance level that is computed by taking the value of the test statistic for the systems and determining how likely a large or larger value could have occurred under the null hypothesis. This probability of the experimental criterion score given the distribution created by the null hypothesis is also known as the p-value.

Statistical testing methods that are commonly used for IR include the sign test, paired Wilcoxon signed rank test, Friedman test, and Student's t-test [141, 278]. In later chapters (except Chapter 6), we use the paired Wilcoxon signed rank test [343], although recent work has indicated some potential issues with this particular test [287]. The null hypothesis of this test is that the results produced by both systems are sampled from the same distribution; in particular that the median difference between pairs of observations is zero. It proceeds as follows. First, it transforms each instance (a pair of observations, i.e., the scores on a retrieval metric for two systems on a particular topic) into absolute values. Then, zero differences are removed and the remaining differences are ranked from lowest to highest. After the signs (that were removed in the first step) are reattributed to the ranks (hence the name signed rank test), the test statistic is calculated. For sample sizes greater than 25, a normal approximation to this statistic exists.

Related to this number is the minimum number of topics one needs to assess to account for the variance in evaluation measures over different topics; 50 topics has been found to be a suitable minimum by Buckley and Voorhees [49], whereas Sanderson and Zobel [278] indicate significant improvements on 25 or less topics does not guarantee that this result will be repeatable on other sets of topics. All of the topic sets we use later in the thesis consist of at least 25 topics, as we describe in the next section.

In the thesis, we look for improvements at the \( p < 0.05 \) level, indicated with a ‘*’. All reported p-values are for two-sided tests. In Chapter 6 we compare multiple methods. There, we use a one-way analysis of variance (ANOVA) test which is a common test when there are more than two systems or methods to be compared. It simultaneously tests for differences in the average score of each method, correcting for the effects of the individual queries. We subsequently use the Tukey-Kramer test to determine which of the individual pairs are significantly different. We use a bold-faced font in the result tables to indicate the best performing model in our result tables.
3. Experimental Methodology

Table 3.2: Statistics of the document collections used in this thesis. \(\mu\) and \(m\) indicate the average and median number of terms in, or concepts assigned to a document respectively and \(\sigma\) the standard deviation. The second group of collections are domain-specific and contain manually assigned concepts as document annotations.

<table>
<thead>
<tr>
<th>Collection</th>
<th>Documents ((\times 10^6))</th>
<th>Size</th>
<th>Terms (\mu)</th>
<th>Terms (\sigma)</th>
<th>Terms (m)</th>
<th>Concepts (\mu)</th>
<th>Concepts (\sigma)</th>
<th>Concepts (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TREC Rob 2004</td>
<td>0.5</td>
<td>2 GB</td>
<td>510</td>
<td>871</td>
<td>359</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>.GOV2</td>
<td>25</td>
<td>426 GB</td>
<td>956</td>
<td>2723</td>
<td>326</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ClueWeb09 cat. A</td>
<td>500</td>
<td>13.4 TB</td>
<td>748</td>
<td>975</td>
<td>460</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ClueWeb09 cat. B</td>
<td>50</td>
<td>1.5 TB</td>
<td>857</td>
<td>1186</td>
<td>507</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CLEF-DS-07/08</td>
<td>0.17</td>
<td>232 MB</td>
<td>62</td>
<td>42</td>
<td>51</td>
<td>10.1</td>
<td>4.2</td>
<td>10</td>
</tr>
<tr>
<td>TREC-GEN-04/05</td>
<td>4.6</td>
<td>20 GB</td>
<td>174</td>
<td>114</td>
<td>171</td>
<td>11.4</td>
<td>5.1</td>
<td>11</td>
</tr>
<tr>
<td>TREC-GEN-06</td>
<td>0.16</td>
<td>12 GB</td>
<td>4160</td>
<td>2750</td>
<td>4525</td>
<td>15.1</td>
<td>6.1</td>
<td>15</td>
</tr>
</tbody>
</table>

3.3 Test Collections

The test collections we employ in this thesis are described in the following sections. We use the Lemur Toolkit for indexing, retrieval, and all language modeling calculations.\(^1\) For all test collections we use only the topic titles as queries. The test collections described first are used for our experiments in Chapters 4 and 7. For all of these collections, we remove a modest list of around 400 stopwords. Our retrieval model presented in Chapter 5 requires collections in which the documents have been manually annotated with an appropriate concept language. The test collections that we describe last (CLEF-DS and TREC-GEN) both satisfy this requirement.

Below we provide a more fine-grained description of each test collection. Tables 3.2, 3.3, and 3.4 list descriptive statistics from each test collection.

3.3.1 TREC Robust 2004

The first is TREC Robust 2004 (TREC-ROB-04), comprising a relatively small document collection and topics which were selected because of their low performance in the TREC ad hoc task [329]. It is the smallest of all collections used in this thesis and contains TREC disks 4 and 5, minus the Congressional Record [329]. The documents are small news articles from the Financial Times, Federal Register, LA Times, and Foreign Broadcast Information Service, covering 1989 through 1996. It is a collection that is routinely used when evaluating the performance of relevance feedback algorithms; 200 of its 250 topics were selected from earlier TREC ad hoc tracks based on their relatively poor performance and the ineffectiveness

\(^1\)See \url{http://sourceforge.net/projects/lemur}.\]
3.3. Test Collections

of relevance feedback techniques; 50 new topics were developed especially for the track.

3.3.2 TREC Terabyte 2004–2006

The second document collection is .GOV2, used in the TREC Terabyte, Million Query, and Relevance Feedback tracks [48, 55]; it contains a crawl of websites from the .gov domain. The TREC Terabyte track ran from 2004 through 2006 and used the first substantially sized TREC document collection [55]; its goal was to develop an evaluation methodology for terabyte-scale document collections. As topic set for this test collection (TREC-TB) we use the combined topics from all years.

3.3.3 TREC Relevance Feedback 2008

This test collection comprises test data provided by the TREC Relevance Feedback track, where the task is to retrieve additional relevant documents given a query and an initial set of relevance assessments [48]. Retrieval is done on the TREC Terabyte collection (the .GOV2 corpus) using 264 topics taken from earlier TREC Terabyte and TREC Million Query tracks [4, 55].

For our explicit relevance feedback experiments (TREC-RF-08) we take the 33 TREC Terabyte topics which were selected from the full set of available topics for an additional round of assessments [48]. A large set of relevance assessments was provided for these topics (159 relevant documents on average, with a minimum of 50 and a maximum of 338). Participating systems were to return 2500 documents, from which the initially provided relevant documents were removed, a procedure similar to residual ranking (when performing residual ranking, all judged documents are removed—instead of only the relevant ones). The resulting rankings were then pooled and re-assessed. This yielded 55 new relevant documents on average per topic, with a minimum of 4 and a maximum of 177. We follow the same setup by keeping only the newly assessed, relevant documents for evaluation and discard all initially judged documents from the final rankings in our experiments.

In order to evaluate pseudo relevance feedback on this test collection (TREC-PRF-08), we use all 264 topics and the combined relevance assessments, i.e., the “original” pools and the new assessments.

3.3.4 TREC Web 2009

The fourth ad hoc test collection that we use has ClueWeb09 as its document collection (TREC-WEB-09). It was employed at the TREC 2009 and 2010 Web Track [72]. It is a large-scale web crawl and contains the largest number of documents. Two subsets are identified; Category B (that contains over 50,000,000
3. Experimental Methodology

### Table 3.3: Statistics of the topic sets used in this thesis.

<table>
<thead>
<tr>
<th>Collection</th>
<th>With(out) rel. docs</th>
<th>Length</th>
<th>Per topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>TREC-ROB-04</td>
<td>249 (1)</td>
<td>2 μ 0.71, σ 1 σ 5</td>
<td>Total 17412, μ 70, Min. 3, Max. 448</td>
</tr>
<tr>
<td>TREC-TB</td>
<td>149 (0)</td>
<td>3 μ 0.88, σ 1 σ 5</td>
<td>Total 26917, μ 180, Min. 4, Max. 617</td>
</tr>
<tr>
<td>TREC-PRF-08</td>
<td>264 (0)</td>
<td>3 μ 1.0, σ 1 σ 8</td>
<td>Total 12639, μ 47, Min. 4, Max. 457</td>
</tr>
<tr>
<td>TREC-RF-08</td>
<td>31 (2)</td>
<td>3 μ 1.0, σ 1 σ 6</td>
<td>Total 1723, μ 55, Min. 4, Max. 177</td>
</tr>
<tr>
<td>TREC-WEB-09</td>
<td>49 (1)</td>
<td>1 μ 0.85, σ 1 σ 4</td>
<td>Total 5684, μ 116, Min. 2, Max. 260</td>
</tr>
<tr>
<td>CLEF-DS-07</td>
<td>25 (0)</td>
<td>4 μ 1.6, σ 2 σ 8</td>
<td>Total 4530, μ 181, Min. 18, Max. 497</td>
</tr>
<tr>
<td>CLEF-DS-08</td>
<td>25 (0)</td>
<td>3 μ 1.7, σ 2 σ 8</td>
<td>Total 2133, μ 85, Min. 4, Max. 206</td>
</tr>
<tr>
<td>TREC-GEN-04</td>
<td>50 (0)</td>
<td>5 μ 3.0, σ 1 σ 16</td>
<td>Total 4584, μ 93, Min. 2, Max. 709</td>
</tr>
<tr>
<td>TREC-GEN-05</td>
<td>49 (1)</td>
<td>5 μ 2.6, σ 2 σ 12</td>
<td>Total 1449, μ 55, Min. 2, Max. 234</td>
</tr>
<tr>
<td>TREC-GEN-06</td>
<td>26 (2)</td>
<td>5 μ 2.5, σ 2 σ 12</td>
<td>Total 8268, μ 165, Min. 1 σ 697</td>
</tr>
<tr>
<td>TREC-WEB-2009 (Cat. A)</td>
<td>5684</td>
<td>116</td>
<td>260</td>
</tr>
<tr>
<td>TREC-WEB-2009 (Cat. B)</td>
<td>4002</td>
<td>82</td>
<td>179</td>
</tr>
</tbody>
</table>

### Table 3.4: Statistics of the relevant documents per collection used in this thesis.

<table>
<thead>
<tr>
<th>Collection</th>
<th>Per topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>TREC-ROB-04</td>
<td>Total 17412, μ 70, Min. 3, Max. 448</td>
</tr>
<tr>
<td>TREC-TB</td>
<td>Total 26917, μ 180, Min. 4, Max. 617</td>
</tr>
<tr>
<td>TREC-PRF-08</td>
<td>Total 12639, μ 47, Min. 4, Max. 457</td>
</tr>
<tr>
<td>TREC-RF-08</td>
<td>Total 1723, μ 55, Min. 4, Max. 177</td>
</tr>
<tr>
<td>TREC-WEB-2009 (Cat. A)</td>
<td>Total 5684, μ 116, Min. 2, Max. 260</td>
</tr>
<tr>
<td>TREC-WEB-2009 (Cat. B)</td>
<td>Total 4002, μ 82, Min. 2, Max. 179</td>
</tr>
<tr>
<td>CLEF-DS-07</td>
<td>Total 4530, μ 181, Min. 18, Max. 497</td>
</tr>
<tr>
<td>CLEF-DS-08</td>
<td>Total 2133, μ 85, Min. 4, Max. 206</td>
</tr>
<tr>
<td>TREC-GEN-04</td>
<td>Total 4584, μ 93, Min. 2, Max. 709</td>
</tr>
<tr>
<td>TREC-GEN-05</td>
<td>Total 1449, μ 55, Min. 2, Max. 234</td>
</tr>
<tr>
<td>TREC-GEN-06</td>
<td>Total 8268, μ 165, Min. 1 σ 697</td>
</tr>
</tbody>
</table>

English web pages) and Category A (that contains over 500,000,000 English web pages). In 2009, participating runs were evaluated using shallow pools and the methodology introduced by the TREC Million Query track [4, 61, 62] as introduced above. The 50 ad hoc topics are taken from a web search engine’s query logs.

#### 3.3.5 CLEF Domain-Specific 2007–2008

The CLEF domain-specific track evaluates retrieval on structured scientific documents, using bibliographic databases from the social sciences domain as document collections [244, 245]. The track emphasizes leveraging the structure of data in collections (defined by concept languages) to improve retrieval perfor-
mance. The 2007 (CLEF-DS-07) and 2008 (CLEF-DS-08) tracks use the combined German Indexing and Retrieval Testdatabase (GIRT) and Cambridge Scientific Abstracts (CSA) databases as their document collection. The GIRT database contains extracts from two databases maintained by the German Social Science Information Centre from the years 1990–2000. The English GIRT collection is a pseudo-parallel corpus to the German GIRT collection, providing translated versions of the German documents (17% of these documents contain an abstract). For the 2007 domain-specific track, an extract from CSA’s Sociological abstracts was added, covering the years 1994, 1995, and 1996. Besides the title and abstract, each CSA record also contains subject-describing keywords from the CSA Thesaurus of Sociological Indexing Terms and classification codes from the Sociological Abstracts classification. In this sub-collection, 94% of the records contains an abstract. We only use the English mono-lingual topics and relevance assessments, which amounts to a total of 50 test topics. The documents in the collection contain three separate fields with concepts, we use CLASSIFICATION-TEXT-EN.

3.3.6 TREC Genomics 2004–2006

The document collection for the TREC 2004 and 2005 Genomics ad hoc search task (TREC-GEN-04 and TREC-GEN-05) consists of a subset of the MEDLINE database [129, 130]. MEDLINE is the bibliographic database maintained by the U.S. National Library of Medicine (NLM). At the time of writing, it contains over 18.5 million biomedical citations from around 5,500 journals and several hundred thousand records are added each year. Despite the growing availability of full-text articles on the Web, MEDLINE remains a central access point for biomedical literature. Each MEDLINE record contains free text fields (such as title and abstract), a number of fields containing other metadata (such as publication date and journal), and, most important for our model in Chapter 5, terms from the MeSH thesaurus. We only use the main descriptors, without qualifiers. MeSH terms are manually assigned to citations by trained annotators from the NLM. The over 20,000 biomedical concepts in MeSH are organized hierarchically, see Figure 1.2 for an example. Relationships between concepts are primarily of the “broader/narrower than” type. The “narrower than” relationship is close to expressing hypernymy (is a), but can also include meronymy (part of) relations. One concept is narrower than another if the documents it is assigned to are contained in the set of documents assigned to the broader term. Each MEDLINE record is annotated with 10–12 MeSH terms on average.

It should be noted that the MeSH thesaurus is not the most appropriate for Genomics information retrieval, since it covers general biomedical concepts rather than the specific genomics terminology used in the TREC topics [305]. Despite this limited coverage, the thesaurus can still be used to improve retrieval effectiveness, as we will show later.
The document collection for TREC Genomics 2004 and 2005 contains 10 years of citations covering 1993 to 2004, which amounts to a total of 4,591,008 documents. All documents have a title, 75.8% contain an abstract and 99% are annotated with MeSH terms. For the 2004 track, 50 test topics are available, with an average length of 7 terms, cf. Table 3.3. The 50 topics for 2005 (one of which has no relevant documents) follow pre-defined templates, so-called Generic Topic Types. An example of such a template is: “Find articles describing the role of [gene] in [disease]”, where the topics instantiate the bold-faced terms. The topics in our experiments are derived from the original topic by only selecting the instantiated terms and discarding the remainder of the template.

The TREC 2006 Genomics track introduced a full-text document collection, replacing the bibliographical abstracts from the previous years [131]. The documents in the collection are full-text versions of scientific journal papers. The files themselves are provided as HTML, including all the journal-specific formatting. Most of the documents (99%) have a valid Pubmed identifier through which the accompanying MEDLINE record can be retrieved. We use the MeSH terms assigned to the corresponding citation as the annotations of the full-text document.

The 2006 test topics are again based on topic templates and instantiated with specific genes, diseases or biological processes. Thus, we preprocess them in a similar fashion as the topics for the TREC Genomics 2005 track, by removing all the template-specific terms. This test collection has 28 topics, of which 2 do not have any relevant documents in the collection. The task put forward for this test collection is to first identify relevant documents and then extract the most relevant passage(s) from each document; relevance is measured at the document, passage, and aspect level. We do not perform any passage extraction and only use the judgments at the document level.\(^1\)

### 3.4 Parameter Settings

Bennett et al. [30] find that the level of smoothing has a significant influence on the resulting retrieval performance and that optimal smoothing parameters are dependent on the query set as well as the collection. They also observe that longer queries require more aggressive smoothing, a finding corroborated by Zhai and Lafferty [355]. In later chapters we need to set values for the smoothing parameter associated with our retrieval model presented in Chapter 2. In particular, we set $\mu$ (cf. Eq. 2.7 on page 16) to the average length of documents in the collection.

\(^1\)2007 was the final year of the TREC Genomics track and used the same document collection as 2006. However, in this edition a new task was introduced and because of the different nature of that task, we do not perform experiments using the 2007 topics.
3.5 Summary

Some of the (pseudo) relevance feedback models in use and under investigation in later chapters require additional parameter settings. The models that we evaluate have the following parameters in common:

- $|V_Q|$ (the number of terms with the highest probability to be included in the query model),
- $|R|$ (the number of feedback documents used), and
- $\lambda_Q$ (the value of the query interpolation factor, cf. Eq. 2.10).

There are various approaches that may be used to estimate these parameters. One can optimize the set of parameters on one test collection and evaluate on the other, use some kind of cross-validation, or designate a set of topics as training topics which are subsequently excluded from the final evaluation. Ideally, we would like to use a form of gradient ascent on the retrieval metric we aim to optimize. None of these measures are continuous, differentiable functions of the set of parameters, however, and many local optima exist [262]. A possible solution is to define another function that does have these properties [54], but typically, a grid or line search is employed to find the optimal values for the parameters, see e.g. [119, 173, 189, 196, 223, 224, 235, 262, 356]. This is also the approach we employ in later chapters. While computationally expensive (exponential in the number of parameters), it does provide us with an upper bound on the retrieval performance that one might achieve using the described models.

3.5 Summary

In this chapter we have introduced our experimental environment, including the relevance assessments, evaluation metrics, significance tests, test collections, and parameter settings. These will serve as the foundation of the experiments upon which we report in later chapters.