Free Software for research in Information Retrieval and Textual Clustering

Emmanuel ECKARD and Jean-Cédric CHAPPELIER

20th April 2007

Abstract

The document provides an overview of the main Free ("Open Source") software of interest for research in Information Retrieval, as well as some background on the context. I provides a guideline for choosing appropriate tools.

Contents

1 Introduction

Natural language processing (NLP) has been an area of enormous development since the 1950s. With the avent of the Internet and the generalisation of computers, information has become so abundant that automatic systems are necessary to manage it. Furthermore, the democratisation of the World Wide Web has created a demand of the general public for Natural language processing services like Information Retrieval (Google, Yahoo,. . .), automated translation (Altavista – Babel Fish Translation), etc.

Textual NLP can be broken into numerous sub-fields, each representing specific problems, and providing specific applications. Some of these fields are strongly related, either because they bear similarities (for instance Information Retrieval and Textual Clustering, which consist in essentially the same processing, applied either on one set of documents, or to documents of two sets), or because one of the fields provide services needed by the other (for instance chunking or tokenising provide facilities used to improve Information Retrieval). As a matter of fact, NLP can only be performed through a series of distinct steps. Depending of the research, focus will be made on one step or another; what is considered to be an input or output will vary; and what is considered to be a "low-level" or "high-level" process will vary, depending on whether a researcher specialises in indexing, tokenising, document clustering, evaluation, etc.

This document will adopt the point of view of Information Retrieval, in which the input is a set of documents in natural language, represented in a computer-friendly form called the *vector space representation*. In this approch, text is assumed to have already been *tokenised* and *lemmatised*, and its indexation is considered to be a low-level task. Other processes operate with inputs and outputs similar to those of Information Retrieval, notably clustering, classification, filtering, machine translation and question answering.

Information Retrieval has become a widespread technique since the popularisation of the World Wide Web. Web-based Information Retrieval has put a particular emphasis on management of huge sets of data, in the terabyte order of magnitude, and quick response to user queries (especially so when one takes network lag into account: in modern web-based Information retrieval engines, network lag amounts to half the time the user waits before getting his answer). The need for efficient retrieval over large amounts of data has given birth to numerous research efforts in academia, and to the building of very large systems in the industry. The recent editions of the TREC conference have held the "Web" and the "Terabyte" tracks, with corresponding corpora made available to researchers.

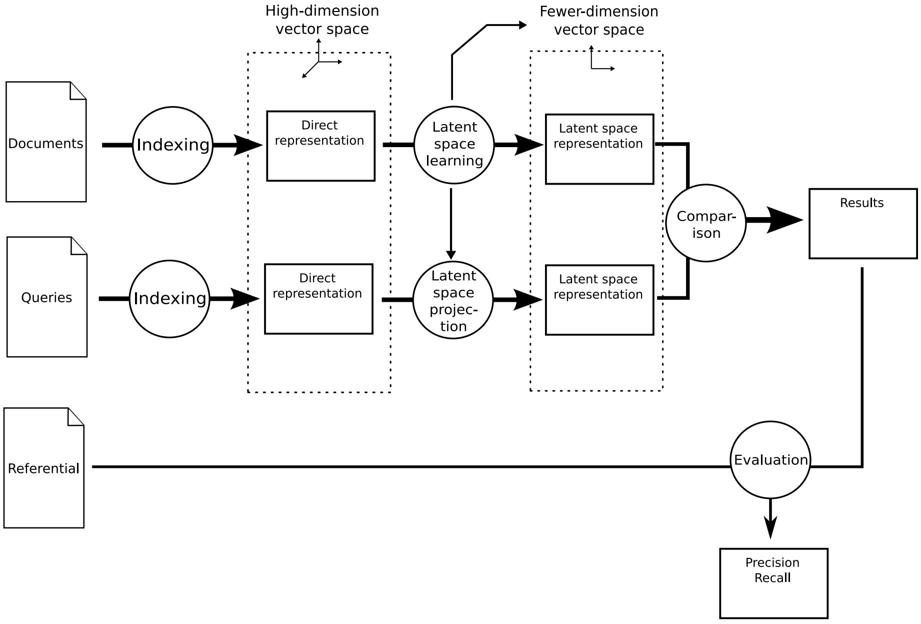

Figure 1: Information retrieval toolchain with Latent Features. Indexing can incorporate steps like chunking, disambiguisation or stemming, and utilise exogenous information. The latent features step is grounded on a model of the parameters underlying to the documents — examples of such models are Naive Bayesian , LSI and PLSI, or Smoothed Dirichlet. The comparison part relies on a similarity appropriate for the vector space by which the documents are represented.

In this example, the Information Retrieval part consists in Latent Space Learning, Latent Space projection and Comparison.

2 Software

2.1 Information retrieval

The situation of Information Retrieval tools is relatively delicate: as of this writing, no software framework is accepted as the standard toolset for community-wide usage. As a matter of fact, some specific laboratories promote their own tools, developped in different programming languages, focusing on different aspects of Information Retrieval, and made available under a variety of different licences. Adopting a software framework is critical for several reasons: efficient access, storage and handling of data is a complex, specific task which requires optimisations falling way beyond the scope of Information Retrieval research; indexing of data might require steps for lemmatisation, chunking, word sense disambiguisation, etc. which usually also falls outside of the scope of the intended research; off-the-shelf software provide complete toolchain allowing to perform experiments with known models, and hence allowing baseline comparisons. Free (or "open-source") software is especially well-suited for these tasks because the absence of charges allows testing several tools; because the availability of the code makes it possible (if not always easy) to implement new algorithms; and because the general availability of the software framework makes it possible for third parties to use applications developed during the research, if the application is suitable.

A general trait apparent in most available software is the effort made to produce applications capable of scaling up to hundreds of Gigabytes of data, in consistency with the Terabyte track of TREC, which explores the behaviour of Information Retrieval models when used on very large quantities of data.

A review of the most proeminent sofware follows. This review was made available to the French Information Retrieval community[1]. Synthetic tables of the results are available as figures 3, 4 and 5 (p.7).

2.1.1 Lemur

Lemur is a toolkit for information retrieval and language modelisation. It is licenced under the BSD licence, and can be obained from `http://www.lemurproject.org/`. The programme is well-maintained, and has been used as a baseline in research [2].

Lemur provides six retrieval models: TFIDF, Okapi (BM 25), Simple KL, InQuery, CORI collection selection, cosine similarity, Indri SQL. Additionally, the optional classification toolkit of Lemur provides an implementation of Hoffman's Probabilistic Semantic Analysis[1] — or at least the Expectation-Maximisation-based learning part of it [2] : as such, queries cannot be processed easily without an operation of *folding in*, which is neither implemented in Lemur, nor very well-defined in the litterature.

Lemur in itself is a library. The package provides a stand-alone Information retrieval engine known as *Indri*. Indri is parallelisable, can be used as a filter,

[1] at `http://www.atala.org/AtalaPedie/index.php?title=Utilisateur:Emmanuel.eckard/Logiciels_d\%27IR` and `http://www.atala.org/AtalaPedie/index.php?title=Utilisateur:Emmanuel.eckard/Lemur-Terrier-Xapian`

[2] We have implemented another version of the learning algorithm on top of Xapian. Our implementation has been tested as giving more accurate results than the Lemur version. See section 2.1.6 and figure 2, p. 6

and scales to the terabyte.

Lemur provides indexers able to read PDF, HTML, XML, and TREC syntax. UTF-8 is supported.

2.1.2 Lucene

Lucene is an Information retrieval library. It is supported by the Apache Software Foundation[3] and is available under the Apache Software Licence from http://lucene.apache.org/java/docs/index.html.

Lucene was written in Java, but can be used with Delphi, Perl, C#, C++, Python, Ruby and PHP.

The LucQE Lucene Query Expansion Module allows using Lucene for TREC experiements[4].

2.1.3 Terrier (TERabyte RetrIEveR)

Terrier is an IR system for large quantities of data. It is written in Java and published under the Mozilla Free licence[5].

Terrier is said to have "full TREC capabilities including the ability to index, query and evaluate the standard TREC collections, such as AP, WSJ, WT10G, .GOV and .GOV2."

Terrier provides tf-idf, Okapi's BM25 and Rocchio's query expansion. It has been tested to scale to all TREC collections.

Developpment tips are given at http://ir.dcs.gla.ac.uk/terrier/doc/terrier_develop.html. Terrier uses a framework application; the user must write an `appmain()`" application. Options are given in XML documents.

2.1.4 Zettair

Zettair is a textual Information retrieval engine published RMIT University under a BSD licence[6].

Zettair allows indexation of text, HTML and TREC formats. A tutorial is available at http://www.seg.rmit.edu.au/zettair/start.html Zettair outputs query logs in the TrecEval format.

Zettair has been tested on 426 GB database of the TREC Terabyte track.

Zettair is written in C.

2.1.5 Zebra

Zebra is an indexation and retrieval engine available under the GPL from http://www.indexdata.dk/zebra/. Zebra was tested as scaling up to dozens of GB.

Zebra is written in C.

[3] Apache is the most widely used HTTP server on the World Wide Web. It is Free software, available under the Apache Software Licence.

[4] http://lucene-qe.sourceforge.net/

[5] Terrier is available from http://ir.dcs.gla.ac.uk/terrier/download.html; `ratheroddlyforFreesoftware,subscriptionisrequired`, and documention, from http://ir.dcs.gla.ac.uk/terrier/documentation.html

[6] available from http://www.seg.rmit.edu.au/zettair/

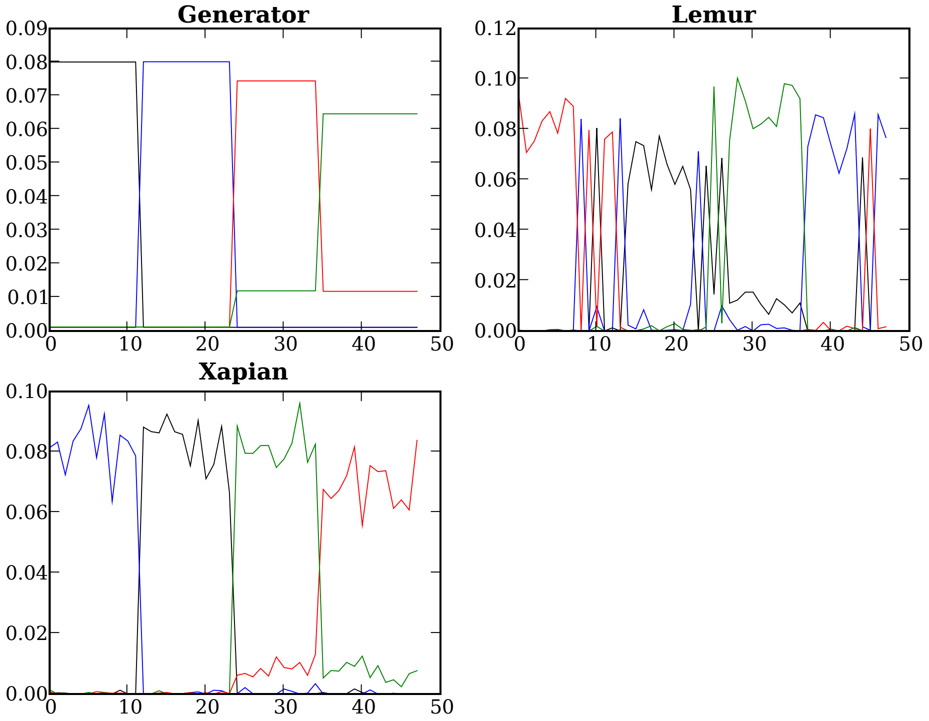

Figure 2: Compared performences of the PLSA learning algorithm as implemented in Lemur, and as implemented by us on Xapian. The "true" distribution used to generate the data is also shown.

2.1.6 Xapian

Xapian is an Information retrieval library focusing on probabilistic retrieval. A stand-alone retrieval engine named Omega is provided. Xapian is available under the GPL from `http://www.xapian.org/`.

Xapian is written in C++, and can be interfaced with Perl, Python, PHP, Java, TCL and C#.

Xapian provides pre-compiled software packages for the main Linux distributions (`rpm` and `deb`).

While Xapian does not natively provide PLSI, we have implemented a version of PLSI as part of a more general software layer. This version was benchmarked against that of Lemur, and was found to yield more accurate results. 50 "documents" were generated from a given probability distribution, and the algorithms were applied to the result to see how they found the original distribution (see figure 2).

General information	Lemur	Terrier	Xapian
Latest revision	21 June 2007 (Lemur Toolkit version 4.5 and Indri version 2.5)	15 June 2007 (Terrier 1.1.0)	4 July 2007 (Xapian 1.0.2)
Focus			Probabilistic retrieval; large databases
Developpers	Carnegie Mellon and University of Massachusetts [7]	University of Glasgow (under Keith van Rijsbergen) [8]	Dr. Martin Porter, BrightStation PLC (originally Cambridge University) [9]
Licence	BSD	Mozilla	GPL
Language	C / C++	Java	C++
Extendable	Library	“Hooks” for custom modules [10]	Library
Indexing capacity			4×10^9 documents or 256 Terabytes[11]

Figure 3: General comparison of Lemur, Terrier and Xapian

Available IR models	Lemur	Terrier	Xapian
Vector Space	Vector Space	?	tf-idf
LSI	No	No	No
PLSI	Classification only	?	Not native
Boolean	?	?	“TradWeight”
Probabilistic	Okapi (BM25)	?	Okapi (BM25)

Figure 4: Comparison of available IR models in Lemur, Terrier and Xapian

Miscellaneous	Lemur	Terrier	Xapian
Executable application	Indri	Terrier	Omega
TREC compatibility	Native	Native	Not native
Packaging	tar.gz ; pre-compiled packages for MS Windows	tar.gz (registration needed)	RPM, DEB, FreeBSD

Figure 5: Practical notes on Lemur, Terrier and Xapian

2.2 Evaluation

2.2.1 Treceval

TREC[12] is an evaluation campaign to study the efficiency of Information Retrieval methods in English, co-sponsored by the National Institute of Standards and Technology (NIST) and US Department of Defence. Its aim is to provide NLP researchers with sample corpora, queries and tools to benchmark retrieval systems, particularly on large test collections. Each of the collections consists of a set of documents, a set of questions, and a corresponding set of reference files ("right answers").

Among the ressources provided by TREC is TrecEval , a text utility to evaluate the efficiency of retrieval programmes, which has become a *de facto* standard among researchers. TrecEval allows evaluation of TREC results using the evaluation procedures of the NIST[13].

Obtaining, compiling and using Treceval : TrecEval can be obtained from Michel Beigbeder's page `http://www.emse.fr/~mbeig/IR/tools.html`. It compiles quite straigtforwardly with a simple `make` in the source directory (further instructions are given the in the `README` file).

TrecEval is used via the command line in the following way:

```
./treceval reference_file IR_programme_output_file
```

For instance, with a TrecEval compiled as above

```
./trec_eval test/qrels.test test/results.test
```

is a valid command in the source directory

Document and queries files:

Reference file: The reference file `reference_file`, which contains the "correct answers" for queries, is relative to a particular document file/queries file couple (a TrecEval user will not need editing these files). A sample of such a file is given as figure 6. In such a file, each line holds a tuple

```
query_id iter doc_id rank
```

Spaces are used as delimitors.

`query_id` : query identification number, a three-digit integer. Tuples are sorted by increasing `query_id`.

`iter` : iteration constant, required, yet ignored, by TrecEval .

`doc_id` : document identification "number" (in fact a string). It is given by the element found between the "DOCNO" XML tags in the corpus (document) file.

`rank` : the relevance of query `query_id` toward document `docno`.

[12] Text REtrieval Conference, `http://trec.nist.gov/`

[13] the US National Institute of Standard and Technology, `http://www.nist.gov/data/nistsd22.htm`

```
1      0        511      1
1      0        513      1
2      0        80       1
2      0        90       1
(...)
```

Figure 6: Extract of a TrecEval reference file

Result file: The result files `IR_programme_output_file` produced by the IR programme being tested contain the rank and similarity of every possible document/query pair obtained by combining the contents of the document and queries files (see figure 6 for example). Each line of the file is of the form

`query_id iter doc_id rank sim run_id`

Spaces are used as delimitors.

`query_id` : query identification number, a three-digit integer. The results are to be sorted by increasing `query_id`.

`iter` : iteration constant, required, yet ignored, by TrecEval .

`doc_id` : document identification "number" (in fact a string). It is given by the element found between the "DOCNO" XML tags in the corpus (document) file.

`rank` : rank, an integer between 0 and 1000. Like `iter`, this value is required in the file format, but ignored by TrecEval .

`sim` : similarity, a "float" floating-point value which gives the numerical value od the mathematical similarity computer for the couple (query, document)

`run_id` : arbitrary name for the run (execution of the programme). This string is printed in the output at runtime, but does not have any influence otherwise.

```
1    0    15    20    0.0197334    0
1    0    1     21    0            0
2    0    71    1     0.213504     0
2    0    68    2     0.158238     0
(...)
```

Figure 7: Example of a TrecEval answer file (answers given by the programme being tested): queries 1 and 2 are compared to documents 15 and 1, and 71 and 68 respectively. The matching of the couple (doc=71, query=2) ranks 1st, with a similarity of 0.213504; that of (doc=15, query=1) is 15th with a similarity of 0.0197334; the couple (1, 1) comes last, since this query and document are orthogonal (similariry 0).

```
$ ./trec_eval test/qrels.test test/results.test
num_q           all     3
num_ret         all     1500
num_rel         all     561
num_rel_ret     all     131
map             all     0.1785
gm_ap           all     0.1051
R-prec          all     0.2174
bpref           all     0.1981
recip_rank      all     0.4064
ircl_prn.0.00   all     0.4665
ircl_prn.0.10   all     0.3884
ircl_prn.0.20   all     0.3186
ircl_prn.0.30   all     0.2732
ircl_prn.0.40   all     0.2666
ircl_prn.0.50   all     0.2184
ircl_prn.0.60   all     0.0822
ircl_prn.0.70   all     0.0348
ircl_prn.0.80   all     0.0312
ircl_prn.0.90   all     0.0312
ircl_prn.1.00   all     0.0312
P5              all     0.2667
P10             all     0.3000
P15             all     0.3111
P20             all     0.3667
P30             all     0.3333
P100            all     0.2467
P200            all     0.1600
P500            all     0.0873
P1000           all     0.0437
```

Figure 8: Example of a TrecEval terminal output. `R-Prec` gives the R-Precision, and `map` gives the Mean Averable Precision. The series of `ircl_prn` are the interpolation precisions at different values of recall; they can be used to draw a Precision-Recall graph. `P5`, `P10` . . . give the precision at 5, 10 etc. retrieved documents.

Output: TrecEval output is given in a terminal, as seen in figure 8. The various

Useful references:

- `http://www.ir.iit.edu/~dagr/cs529/files/project_files/trec_eval_desc.htm`
- `http://www.cs.colorado.edu/~martin/Csci7000/using_trec_eval.txt`

3 Resources

3.1 Ontologies

Ontologies are hierarchically organised dictionnaries which provide real-worlds knowledge about a set of subjects. Subjects are linked with formalised relations as to provide context for concepts.

3.1.1 Wordnet

WordNet is a multilingual semantic lexicon. While the English version is under a BSD licence and is easily available[14], the multilingual version, EuroWordNet, is a proprietary and commercial project. Eurowordnet supports Dutch, English, Italian, Spanish, German, French, Czech and Estonian.

3.1.2 EDR

Electronic Dictionary Research[15] is a set of "subdictionnaries" comprising lexical dictionaries for Japanese and English, with thesaurus-like concept classifications and corpus databases. EDR is provided on CD-ROMs, for a fee of 50 000 yen per subdictionary. Available subdictionnaries are

- Japanese Word Dictionary (JWD-V030)
- English Word Dictionary (EWD-V030)
- Concept Dictionary (CPD-V030)
- Japanese-English bilingual Dictionary (JEB-V030)
- English-Japanese bilingual Dictionary (EJB-V030)
- Japanese Co-occurrence Dictionary (JCO-V030E and JCO-V030S)
- English Co-occurrence Dictionary (ECC-V030, ECO-V030E and ECO-V030S)
- Technical Terminology Dictionary (TED-V030)

[14] Packages exist for Linux distributions.

[15] `http://www2.nict.go.jp/r/r312/EDR/index.html`

3.2 Document collections

3.2.1 SMART bases

The SMART collections are a set of six articles, queries and reference files triads, available for no charge from `ftp://ftp.cs.cornell.edu/pub/smart/`: ADI, CACM, CISI, CRAN, MED and TIME.

These bases are provided with sets of queries, and relevance lists in the TrecEval format.

They have been largely used to test the capabilities of retrieval systems.

	ADI	CACM	CISI	CRAN	MED	TIME
Terms	2402	15027	16067	12029	20177	35619
Documents	82	3204	1460	1400	1033	425
Queries	35	64	112	225	30	83

3.2.2 TREC

The Trec AP collection[16] is a text retrieval annotated corpus. It is constitued of 242 918 news stories published by the Associated Press in 1988, 1989 and 1990. The data is provided on several CDs available from TREC for researchers only[17] A version of this collection also exists for text categorisation.

3.2.3 Reuters 21578

The Reuters 21578 collection[18] is a frequently-used text categorisation annotated corpus. In is constitued of 21578 Reuters news stories published in 1987. The data is provided in 22 files (21 1000-document file and the last file with the 578 remaining documents). An archive can be downloaded easily[19] as a 8.2 MB tarball file (28.0 MB uncompressed).

Reuters 21578 has its own SGML syntax. It comes with 5 sets of categories: “Exchanges”, “Orgs”, “People”, “Places” and “Topics”.

Set of categories	Exchanges	Orgs	People	Places	Topics
Number of categories	39	56	267	175	135

4 Conclusion

The increase in interest cast on practical applications of Natural Language Processing, the development of computing ressources and the growth of the Free Software movement has contributed to the rise of numerous ressources for Natural Language Processing. From a researcher’s point of view, some specific applications are associated with official or de facto standards (programmes or ressources), while for some important fields the researcher is on his own in a jungle of competiting software.

The TREC conference has set its own standards input and output for Information retrieval and classification modules. In some measure it has succeeded in

[16] `http://www.daviddlewis.com/resources/testcollections/trecap/`

[17] nd for a substancial fee

[18] `http://www.daviddlewis.com/resources/testcollections/reuters21578/`

[19] from `http://www.daviddlewis.com/resources/testcollections/reuters21578/reuters21578.tar.gz`

becomming a de facto standard, notably because of the availability of large annotated databases in the TREC format and of TrecEval . However, TREC does not provide a standard software framework for holding textual data and developping Information Retrieval programmes (or similarly high-level applications), and as of this writing, the task of choosing an appropriate software environment must be undertaken before begining experimentation[20]. Some researchers undergo the process of implementing data structure themselves, which should be strongly discouraged as tedious, frustrating when the core of the research is of a higher level, and producing highly sub-optimal results. Off-the-shelf software can implement compatibility with TREC more or less toroughly, depending on the emphasis casted on reseach or industrial applications. In the general case, a researcher should evaluate the main software available in terms of performance, ease of development and fitness to his purpose, and adapt it with specific and limited ad-hoc code.

[20] By contrast, the CERN provides standard programmes for data analysis in Particle Physics, so the step of choosing software is trivial.

A Description of a few evaluation measures of TrecEval II

Abstract

This section was written by Florian SEYDOUX, and translated by Emmanuel ECKARD. It details the evaluation measures notably used by TrecEval

A.1 Introduction

`TrecEval` is an evaluation tool with a range of measures of *precision* and *recall* to quantify the performance of an information retrieval system. For a set of *queries*, it compares the document index returned by the IR system to an index of relevant documents, called *referential*.

A.1.1 Conventions

Glossary

query: Question submited to the system

system: Information Retrieval system being evaluated. Used as an adjectif, something related to the system.

referential: "truely" relevant elements. Used as an adjectif, something related to the referential.

Mathematical notation

D Set of documents in a document collection (or "corpus"). $|D|$is the cardinality of D(number of documents);

Q Set of the queries asked on the document collection. $|Q|$is the cardinality of Q(number of queries);

$Relev_{t\in Q,D}^{ref}$ Sub-set of documents of D"truely" relevant to a query q; $Relev_{i,s}^{ref}$ given by the referential. We suppose that an ordering relation is defined on the set, denoting how relevant a document is to a query (for instance the most relevant documents have a higher score).

$Relev_{t\in Q,D}^{sys}$ Sub-set of documents of Dreturned by the system when issued a query q. Similarly, an ordering relation is defined on the set.

$\mathcal{P}(d)$ Measure of the *precision* up to document d, defined as

$$\mathcal{P}(d) = \frac{\text{Number of relevant documents extracted up to } d}{\text{Number of documents extracted up to } d}$$

$\mathcal{R}(d)$ Measure of the *recall* up to document d, defined as

$$\mathcal{R}(d) = \frac{\text{Number of relevant documents extracted up to } d}{\text{Number of relevant documents to find}}$$

A.2 Usage

`TrecEval II` may be used by issuing the following command:

```
treceval [-q] [-a] referential system
```

where *referential* is a file listing the relevant documents, and *system* is a file listing the documents returned by the system.

-q option `[-q]` gives the measured valued for each query independentally, instead of the means over all the queries.

-a option `[-a]` prints out additional measures, including measures used in TREC-a but not in TREC-2 and measures considered for future evaluation campains. It is advisable to use this option.

A.2.1 Format of the referential file

The referential file defines the set of the documents which are "truely" relevant for every query. More precisely, it gives the query matching each relevant document. These files are created manually by experts.

The file consists in four columns separated with whitespaces. The lines are ordered by query number, and define [query – document] matches. The format also allows additional information to be set, but this is not used by `TrecEval II`.

The columns give the following information:

query indexing number (integer) of the query of the [query – document] match.

iteration (unused) identifier of the iteration (string of characters). Often set to "0".

document identifier of the document in the [query – document] match (string of characters).

relevance degree of relevance. There are 5 values are valid in the format, but only the first one is used in `TrecEval II`:

1 documents exactly matching the query
2 documents with a high relevance
3 documents relevant to some aspects of the query
4 documents with slight relevance, or included for historical reasons
5 documents with no relevance

Every line must end with a line break, including the last one. The End Of File character follows a line break. See figure 9 for an example.

A.2.2 Format of the system file

The system file defines for each query the set of documents returned by the system being evaluated — that is the documents that the system finds relevant.

```
1   0   document_10                          1
1   0   doc-11                               1
1   0   12                                   1
1   0   article_13                           1
1   0   14                                   1
1   0   15                                   1
2   0   book_23                              1
2   0   22                                   1
2   0   21                                   1
2   0   Twenty_Thousand_Leagues_Under_the_Sea    1
3   0   30                                   1
3   0   32                                   1
3   0   31                                   1
3   0   33                                   1
3   0   34                                   1
3   0   35                                   1
```

Figure 9: Example of a TrecEval referential file, with four columns describing, respectively: the query number (int); the iteration (string; unused); the document identifier (string); and the type of relevance (always 1 in ***TrecEval II***). The EOF follows a linebreak.

More precisely, each document returned by the system is associated with the query to which it is matched, and with the degree of relevance.

The file consists in 6 columns whose lines are ordered by query number. The columns give the following information:

query indexing number (integer) of the query of the [query – document] match.

iteration (unused) identifier of the iteration (string of characters). Often set to “0”.

document identifier of the document in the [query – document] match (string of characters).

ranking ranking of the document in the list of documents returned for a given query. A document with ranking n is the n-th most relevant for the query (integer).

score Measure of the relevance of the [query – document] match (real number). A high value of the score usually denotes a good [query – document] match, and thus a low ranking value of the document for that particular query.

process (unused) identifier of the process (string of characters). Often set to “0”.

Chaque ligne décrivant une association (et en particulier la dernière) doit impérativement être terminée par un saut de ligne (la marque de fin de fichier devant se trouver après ce saut)

Every line must end with a line break, including the last one. The End Of File character follows a line break. See figure 10 for an example.

```
1    0    doc-11                                    1    85    0
1    0    11                                        2    80    0
1    0    document_10                               3    79    0
1    0    15                                        4    78    0
1    0    article_13                                5    76    0
2    0    4                                         1    0.74  0
2    0    23                                        2    0.99  0
2    0    22                                        3    0.85  0
2    0    Twenty_Thousand_Leagues_Under_the_Sea     4    0.50  0
2    0    32                                        5    0.49  0
2    0    20                                        6    0.30  0
3    0    1                                         1    0.99  0
3    0    article_13                                2    0.89  0
3    0    31                                        3    0.80  0
3    0    40                                        4    0.80  0
3    0    34                                        5    0.68  0
3    0    4                                         6    0.58  0
3    0    35                                        7    0.01  0
```

Figure 10: Example of a TrecEval system file, with six columns describing, respectively: the query number (int); the iteration (string; unused); the document identifier (string); the ranking (int); the relevance score (real); and the process identifier (string). The EOF follows a linebreak.

A.3 Available measures

A.3.1 Run example

Issuing the command

```
treceval -a referentiel system
```

will yield the following output:

```
Queryid (Num):        1
Total number of documents over all queries
    Retrieved:        5
    Relevant:         6
    Rel_ret:          3
Interpolated Recall - Precision Averages:
    at 0.00       0.6000
    at 0.10       0.6000
    at 0.20       0.6000
    at 0.30       0.6000
    at 0.40       0.6000
    at 0.50       0.6000
    at 0.60       0.0000
    at 0.70       0.0000
    at 0.80       0.0000
    at 0.90       0.0000
    at 1.00       0.0000
Average precision (non-interpolated) over all rel docs
                  0.2667
Precision:
  At    5 docs:   0.6000
  At   10 docs:   0.3000
  At   15 docs:   0.2000
  At   20 docs:   0.1500
  At   30 docs:   0.1000
  At  100 docs:   0.0300
  At  200 docs:   0.0150
  At  500 docs:   0.0060
  At 1000 docs:   0.0030
R-Precision (precision after R (= num_rel for a query) docs retrieved):
    Exact:        0.5000

----------------------------------------------------------------
The following measures included for TREC 1 compatability

Precision:
   Exact:         0.6000
Recall:
   Exact:         0.5000
   at   5 docs:   0.5000
   at  10 docs:   0.5000
   at  15 docs:   0.5000
   at  20 docs:   0.5000
   at  30 docs:   0.5000
   at 100 docs:   0.5000
   at 200 docs:   0.5000
   at 500 docs:   0.5000
   at 1000 docs:   0.5000
Average interpolated precision for all 11 recall points
   11-pt Avg:     0.3273
Average interpolated precision for 3 intermediate points (0.20, 0.50, 0.80)
    3-pt Avg:     0.4000

----------------------------------------------------------------
The following measures are possible for future TRECs

R-based-Precision (precision after given multiple of R docs retrieved):
    Exact:        0.5000
    At 0.20  R:   0.5000
    At 0.40  R:   0.3333
    At 0.60  R:   0.5000
    At 0.80  R:   0.6000
    At 1.00  R:   0.5000
    At 1.20  R:   0.3750
    At 1.40  R:   0.3333
    At 1.60  R:   0.3000
    At 1.80  R:   0.2727
    At 2.00  R:   0.2500
Relative Precision:
   Exact:         0.6000
   At   5 docs:   0.6000
   At  10 docs:   0.5000
   At  15 docs:   0.5000
   At  20 docs:   0.5000
   At  30 docs:   0.5000
   At 100 docs:   0.5000
   At 200 docs:   0.5000
   At 500 docs:   0.5000
   At 1000 docs:   0.5000
Average precision for first R docs retrieved:
                  0.4222
Fallout - Recall Averages (recall after X nonrel docs retrieved):
    At   0 docs:   0.0000
    At  14 docs:   0.5000
    At  28 docs:   0.5000
    At  42 docs:   0.5000
    At  56 docs:   0.5000
    At  71 docs:   0.5000
    At  85 docs:   0.5000
    At  99 docs:   0.5000
    At 113 docs:   0.5000
    At 127 docs:   0.5000
    At 142 docs:   0.5000
Average recall for first 142 nonrel docs retrieved:
                   0.4941

----------------------------------------------------------------
The following measures are interpolated versions of measures above.
For the following, interpolated_prec(X) == MAX (prec(Y)) for all Y >= X
All these measures are experimental

Average interpolated precision over all rel docs
```

```
                  0.3000
R-based-interpolated-Precision:
    Exact:        0.5000
    At 0.20  R:   0.6000
    At 0.40  R:   0.6000
    At 0.60  R:   0.6000
    At 0.80  R:   0.6000
    At 1.00  R:   0.5000
    At 1.20  R:   0.3750
    At 1.40  R:   0.3333
    At 1.60  R:   0.3000
    At 1.80  R:   0.2727
    At 2.00  R:   0.2500
Average interpolated precision for first R docs retrieved:
                  0.6000

Queryid (Num):        2
Total number of documents over all queries
    Retrieved:        6
    Relevant:         4
    Rel_ret:          3
Interpolated Recall - Precision Averages:
    at 0.00       1.0000
    at 0.10       1.0000
    at 0.20       1.0000
    at 0.30       1.0000
    at 0.40       1.0000
    at 0.50       1.0000
    at 0.60       0.5000
    at 0.70       0.5000
    at 0.80       0.0000
    at 0.90       0.0000
    at 1.00       0.0000
Average precision (non-interpolated) over all rel docs
                  0.6250
Precision:
  At    5 docs:   0.4000
  At   10 docs:   0.3000
  At   15 docs:   0.2000
  At   20 docs:   0.1500
  At   30 docs:   0.1000
  At  100 docs:   0.0300
  At  200 docs:   0.0150
  At  500 docs:   0.0060
  At 1000 docs:   0.0030
R-Precision (precision after R (= num_rel for a query) docs retrieved):
    Exact:        0.5000

----------------------------------------------------------------
The following measures included for TREC 1 compatability

Precision:
   Exact:         0.5000
Recall:
   Exact:         0.7500
   at   5 docs:   0.5000
   at  10 docs:   0.7500
   at  15 docs:   0.7500
   at  20 docs:   0.7500
   at  30 docs:   0.7500
   at 100 docs:   0.7500
   at 200 docs:   0.7500
   at 500 docs:   0.7500
   at 1000 docs:   0.7500
Average interpolated precision for all 11 recall points
   11-pt Avg:     0.6364
Average interpolated precision for 3 intermediate points (0.20, 0.50, 0.80)
    3-pt Avg:     0.6667

----------------------------------------------------------------
The following measures are possible for future TRECs

R-based-Precision (precision after given multiple of R docs retrieved):
    Exact:        0.5000
    At 0.20  R:   1.0000
    At 0.40  R:   1.0000
    At 0.60  R:   0.6667
    At 0.80  R:   0.5000
    At 1.00  R:   0.5000
    At 1.20  R:   0.4000
    At 1.40  R:   0.5000
    At 1.60  R:   0.4286
    At 1.80  R:   0.3750
    At 2.00  R:   0.3750
Relative Precision:
   Exact:         0.7500
   At   5 docs:   0.5000
   At  10 docs:   0.7500
   At  15 docs:   0.7500
   At  20 docs:   0.7500
   At  30 docs:   0.7500
   At 100 docs:   0.7500
   At 200 docs:   0.7500
   At 500 docs:   0.7500
   At 1000 docs:   0.7500
Average precision for first R docs retrieved:
                  0.6667
Fallout - Recall Averages (recall after X nonrel docs retrieved):
    At   0 docs:   0.5000
    At  14 docs:   0.7500
    At  28 docs:   0.7500
    At  42 docs:   0.7500
    At  56 docs:   0.7500
    At  71 docs:   0.7500
    At  85 docs:   0.7500
    At  99 docs:   0.7500
    At 113 docs:   0.7500
    At 127 docs:   0.7500
    At 142 docs:   0.7500
Average recall for first 142 nonrel docs retrieved:
                   0.7447

-----------------------------------------------------------------
The following measures are interpolated versions of measures above.
For the following, interpolated_prec(X) == MAX (prec(Y)) for all Y >= X
All these measures are experimental

Average interpolated precision over all rel docs
                  0.6250
R-based-interpolated-Precision:
    Exact:        0.5000
    At 0.20  R:   1.0000
    At 0.40  R:   1.0000
    At 0.60  R:   0.6667
    At 0.80  R:   0.5000
    At 1.00  R:   0.5000
    At 1.20  R:   0.5000
    At 1.40  R:   0.5000
    At 1.60  R:   0.4286
    At 1.80  R:   0.3750
    At 2.00  R:   0.3750
Average interpolated precision for first R docs retrieved:
                  0.6667

Queryid (Num):        3
Total number of documents over all queries
    Retrieved:        7
    Relevant:         6
    Rel_ret:          4
Interpolated Recall - Precision Averages:
    at 0.00       0.6000
    at 0.10       0.6000
    at 0.20       0.6000
    at 0.30       0.6000
    at 0.40       0.6000
    at 0.50       0.6000
    at 0.60       0.5714
    at 0.70       0.0000
    at 0.80       0.0000
    at 0.90       0.0000
    at 1.00       0.0000
Average precision (non-interpolated) over all rel docs
                  0.3619
Precision:
  At    5 docs:   0.6000
  At   10 docs:   0.4000
  At   15 docs:   0.2667
  At   20 docs:   0.2000
  At   30 docs:   0.1333
  At  100 docs:   0.0400
  At  200 docs:   0.0200
  At  500 docs:   0.0080
  At 1000 docs:   0.0040
R-Precision (precision after R (= num_rel for a query) docs retrieved):
    Exact:        0.5000

----------------------------------------------------------------
The following measures included for TREC 1 compatability

Precision:
   Exact:         0.5714
Recall:
   Exact:         0.6667
   at   5 docs:   0.5000
   at  10 docs:   0.6667
   at  15 docs:   0.6667
   at  20 docs:   0.6667
   at  30 docs:   0.6667
   at 100 docs:   0.6667
   at 200 docs:   0.6667
   at 500 docs:   0.6667
   at 1000 docs:   0.6667
Average interpolated precision for all 11 recall points
   11-pt Avg:     0.3792
Average interpolated precision for 3 intermediate points (0.20, 0.50, 0.80)
    3-pt Avg:     0.4000

----------------------------------------------------------------
The following measures are possible for future TRECs

R-based-Precision (precision after given multiple of R docs retrieved):
    Exact:        0.5000
    At 0.20  R:   0.5000
    At 0.40  R:   0.3333
    At 0.60  R:   0.5000
    At 0.80  R:   0.6000
    At 1.00  R:   0.5000
    At 1.20  R:   0.5000
    At 1.40  R:   0.4444
    At 1.60  R:   0.4000
    At 1.80  R:   0.3636
    At 2.00  R:   0.3333
Relative Precision:
   Exact:         0.6667
   At   5 docs:   0.6000
   At  10 docs:   0.6667
   At  15 docs:   0.6667
   At  20 docs:   0.6667
   At  30 docs:   0.6667
   At 100 docs:   0.6667
   At 200 docs:   0.6667
   At 500 docs:   0.6667
   At 1000 docs:   0.6667
Average precision for first R docs retrieved:
                  0.3222
Fallout - Recall Averages (recall after X nonrel docs retrieved):
    At   0 docs:   0.0000
    At  14 docs:   0.6667
    At  28 docs:   0.6667
    At  42 docs:   0.6667
    At  56 docs:   0.6667
    At  71 docs:   0.6667
    At  85 docs:   0.6667
    At  99 docs:   0.6667
    At 113 docs:   0.6667
    At 127 docs:   0.6667
    At 142 docs:   0.6667
Average recall for first 142 nonrel docs retrieved:
                   0.6573

----------------------------------------------------------------
The following measures are interpolated versions of measures above.
For the following, interpolated_prec(X) == MAX (prec(Y)) for all Y >= X
All these measures are experimental

Average interpolated precision over all rel docs
                  0.3952
R-based-interpolated-Precision:
    Exact:        0.5714
    At 0.20  R:   0.6000
    At 0.40  R:   0.6000
    At 0.60  R:   0.6000
    At 0.80  R:   0.6000
    At 1.00  R:   0.5714
    At 1.20  R:   0.5000
    At 1.40  R:   0.4444
    At 1.60  R:   0.4000
    At 1.80  R:   0.3636
    At 2.00  R:   0.3333
Average interpolated precision for first R docs retrieved:
                  0.5000

Queryid (Num):        3
Total number of documents over all queries
    Retrieved:       18
    Relevant:        16
    Rel_ret:         10
Interpolated Recall - Precision Averages:
    at 0.00       0.7333
    at 0.10       0.7333
    at 0.20       0.7333
    at 0.30       0.7333
    at 0.40       0.7333
    at 0.50       0.7333
    at 0.60       0.3571
    at 0.70       0.1667
    at 0.80       0.0000
    at 0.90       0.0000
    at 1.00       0.0000
Average precision (non-interpolated) over all rel docs
                  0.4179
Precision:
  At    5 docs:   0.5333
  At   10 docs:   0.3333
  At   15 docs:   0.2222
  At   20 docs:   0.1667
  At   30 docs:   0.1111
  At  100 docs:   0.0333
  At  200 docs:   0.0167
  At  500 docs:   0.0067
  At 1000 docs:   0.0033
R-Precision (precision after R (= num_rel for a query) docs retrieved):
    Exact:        0.5000

----------------------------------------------------------------
The following measures included for TREC 1 compatability

Precision:
   Exact:         0.5571
Recall:
   Exact:         0.6389
   at   5 docs:   0.5000
   at  10 docs:   0.6389
   at  15 docs:   0.6389
   at  20 docs:   0.6389
   at  30 docs:   0.6389
   at 100 docs:   0.6389
   at 200 docs:   0.6389
   at 500 docs:   0.6389
   at 1000 docs:   0.6389
Average interpolated precision for all 11 recall points
   11-pt Avg:     0.4476
Average interpolated precision for 3 intermediate points (0.20, 0.50, 0.80)
    3-pt Avg:     0.4889

-----------------------------------------------------------------
The following measures are possible for future TRECs

R-based-Precision (precision after given multiple of R docs retrieved):
    Exact:        0.5000
    At 0.20  R:   0.6667
    At 0.40  R:   0.5556
    At 0.60  R:   0.5556
    At 0.80  R:   0.5667
    At 1.00  R:   0.5000
    At 1.20  R:   0.4250
    At 1.40  R:   0.4259
    At 1.60  R:   0.3762
    At 1.80  R:   0.3371
    At 2.00  R:   0.3194
Relative Precision:
   Exact:         0.6722
   At   5 docs:   0.5667
   At  10 docs:   0.6389
   At  15 docs:   0.6389
   At  20 docs:   0.6389
   At  30 docs:   0.6389
   At 100 docs:   0.6389
   At 200 docs:   0.6389
   At 500 docs:   0.6389
   At 1000 docs:   0.6389
Average precision for first R docs retrieved:
                  0.4704
Fallout - Recall Averages (recall after X nonrel docs retrieved):
    At   0 docs:   0.1667
    At  14 docs:   0.6389
    At  28 docs:   0.6389
    At  42 docs:   0.6389
    At  56 docs:   0.6389
    At  71 docs:   0.6389
    At  85 docs:   0.6389
    At  99 docs:   0.6389
    At 113 docs:   0.6389
    At 127 docs:   0.6389
    At 142 docs:   0.6389
Average recall for first 142 nonrel docs retrieved:
                   0.6320

-----------------------------------------------------------------
The following measures are interpolated versions of measures above.
For the following, interpolated_prec(X) == MAX (prec(Y)) for all Y >= X
All these measures are experimental

Average interpolated precision over all rel docs
                  0.4401
R-based-interpolated-Precision:
    Exact:        0.5238
    At 0.20  R:   0.7333
    At 0.40  R:   0.7333
    At 0.60  R:   0.6222
    At 0.80  R:   0.5667
    At 1.00  R:   0.5238
    At 1.20  R:   0.4583
    At 1.40  R:   0.4259
    At 1.60  R:   0.3762
    At 1.80  R:   0.3371
    At 2.00  R:   0.3194
Average interpolated precision for first R docs retrieved:
                  0.5889
```

A.3.2 Description of the measures

The standard measures available with *`TrecEval II`* are :

Queryid Number of queries in the referential (i.e. $|Q|$);

Total number of doc. over all queries Every of the following sums is determined over all queries:

- **Retrieved** total numer of documents found by the system being evaluated, i.e. $\sum_{q \in Q} |Relev^{sys}_{q,D}|$, or $\sum_{q \in Q} |Relev^{sys}_{q,D}|$
- **Relevant** total number of relevant documents, i.e. $\sum_{q \in Q} |Relev^{ref}_{q,D}|$, or $\sum_{q \in Q} |Relev^{ref}_{q,D}|$
- **Rel_ret** total number of relevant documents found by the system, i.e. $\sum_{q \in Q} |Relev^{sys}_{q,D} \bigcap Relev^{ref}_{q,D}|$

Interpolated Recall - Precision Averages (at α) The value given for a given recall α is the mean over all queries of the maximum precision over the relevant documents found by the system ($Relev^{ok}$) with a recall equal or superior to α. i.e.

$$\overline{P_\alpha} = \frac{1}{|Q|} \sum_{q \in Q} \max_{d \in Relev^{ok}_{q,D} | \mathcal{R}(d) \geqslant \alpha} \big(\mathcal{P}(d) \big)$$

For the recall value $\alpha = 0$, the precision is the mean of the maximum precisions for all the queries.

Average precision (non-interpolated) over all rel docs Mean over all queries of the mean precision for $Relev^{ref}$, i.e.

$$\overline{P^{ref}} = \frac{1}{|Q|} \sum_{q \in Q} \left(\frac{1}{|Relev^{ref}_{q,D}|} \sum_{d \in Relev^{ok}_{q,D}} \big(\mathcal{P}(d) \big) \right)$$

Precision (at α doc) Mean over all queries of the precision obtained after extracting the α^{th} document retrieved by the system, i.e. the number of relevant documents retrieved after retrieving α documents, divided by α. Missing documents (beyond α) are considered to be irrelevant.

R-Precision Mean over all queries of the precision obtained after retrieving as many documents as there are relevant documents — i.e. precision at $\alpha = |Relev^{ref}_q|$).

A.3.3 Additional measures from `Trec I`:

the additional measures of the initial campaign *`Trec I`*, available with option `-a`, are:

Precision exact Mean over all queries of the definitive precision, i.e. precision obtained after retrieval of all documents found by the system (Precision at $\alpha = |Relev^{sys}_q|$).

Recall Mean over all queries of recall, after retrieval of α documents; `recall exact` is the recall obtained after retrieving all documents in $Relev^{sys}$.

Average interpolated 11-points precision Mean of the precisions on the 11 points of recall (0.0, 0.1, 0.2, ... 0.9, 1.0)

Average interpolated 3-points precision Mean of the precisions on the 3 points of recall 0.2, 0.5 and 0.8.

A.3.4 Additional measures for future campaigns:

the additional measures of future campaigns, available with option `-a`, are the means over all queries of:

R-based-Precision Precision obtained after retrieving $\lceil \lambda \times |Relev_q^{ref}| \rceil$ document; for $\lambda = 1$, the `R-based-Precision` is "exact" and amounts to the R-Precision

Relative Precision Exact if $|Relev_q^{ref}| > |Relev_q^{sys}|$, the Relative Precision amounts to the precision after retrieving all documents found by the system. Else, it amounts to the recall at this point.

at α doc if $\alpha < |Relev_q^{ref}|$, the Relative Precision amounts to the precision after retrieving α documents found by the system. Else, it amounts to the recall at this point.

Average precision for first R doc retrieved if $|Relev_q^{sys}| < |Relev_q^{ref}|$, the Average precision for first R doc retrieved has the value

$$A = \frac{1}{|Relev_q^{ref}|} \cdot \left(\sum_{d \in Relev_q^{sys}} \mathcal{P}(d) \right) + \left(\sum_{i=|Relev_q^{sys}|}^{|Relev_q^{ref}|-1} \frac{|Relev_q^{ok}|}{i} \right)$$

else, it has the value

$$A = \frac{1}{|Relev_q^{ref}|} \cdot \sum_{d \in \langle Relev_q^{sys} \rangle_{\lll |Relev_q^{ref}|-1}} \mathcal{P}(d)$$

with $\langle E \rangle_{\lll k}$ being set E restricted to its k first elements.

Fallout - Recall Average (at α doc) Recall before retrieval of the $\alpha + 1^{\text{th}}$ irrelevant document($Relev^{err}$ [21])

Average recall for first k nonrel doc retrieved mean of the recall over the $k = 142$ first irrelevant documents, i.e.

$$= \frac{1}{k} \cdot \left(\sum_{d \in \langle Relev_q^{err} \rangle_{\lll k}} \mathcal{R}(d) \right) \cdot \left(\sum_{i=|Relev_q^{err}|}^{k} \frac{|Relev_q^{ok}|}{|Relev_q^{ref}|} \right)$$

[21] with $Relev^{err} = Relev^{sys} \setminus Relev^{ok}$

A.3.5 Interpolated precisions:

the values of the precisions interpolated against the maximum of the precisions obtained with the documents still to be retrieved[22], are the means over all queries of:

Average interpolated precision over all rel docs average interpolated precision obtained on $Relev_q^{ref}$, i.e.

$$= \frac{1}{|Relev_q^{ref}|} \cdot \sum_{d \in Relev_q^{ok}} \max_{d' \in \langle d^{+*} \rangle} \left(\mathcal{P}\left(d'\right) \right)$$

where $\langle d^{+*} \rangle$ is the set of all documents after d.

R-based-interpolated-Precision Interpolated Precision obtained after retrieval of $\lceil \lambda \times |Relev_t^{ref}| \rceil$ documents.

Average interpolated precision for first R doc retrieved If $|Relev_q^{sys}| < |Relev_q^{ref}|$, Average interpolated precision for first R doc retrieved amounts to

$$A = \frac{1}{|Relev_q^{ref}|} \cdot \left(\sum_{d \in Relev_q^{sys}} \max_{d' \in \langle d^{+*} \rangle} \mathcal{P}\left(d'\right) \right) + \left(\sum_{i=|Relev_q^{sys}|}^{|Relev_q^{ref}|-1} \frac{|Relev_q^{ok}|}{i} \right)$$

else,

$$A = \frac{1}{|Relev_q^{ref}|} \cdot \sum_{d \in \langle Relev_q^{sys} \rangle_{\lll |Relev_q^{ref}|-1}} \max_{d' \in \langle d^{+*} \in Relev_q^{sys} \rangle} \mathcal{P}\left(d'\right)$$

[22] said to be “experimental” measures

References

[1] T. Hofmann. Learning the similarity of documents: An information-geometric approach to document retrieval and categorization. In S. A. Solla, T.K. Leen, and K.-R. Müller, editors, *Proc. of Advances in Neural Information Processing Systems 12 (NIPS'99)*, pages 914–920. MIT Press, 2000.

[2] Ramesh Nallapati. *The Smoothed Dirichlet Distribution: Understanding Cross-Entropy Ranking in Information Retrieval*. PhD thesis, University of Massachusetts, Amherst, MA, USA, 2006.