Machine Learning for Information Retrieval: TREC 2009 Web, Relevance Feedback and Legal Tracks

Gordon V. Cormack and Mona Mojdeh
Cheriton School of Computer Science
University of Waterloo

1 Overview

For the TREC 2009, we exhaustively classified every document in each corpus, using machine learning methods that had previously been shown to work well for email spam [9, 3]. We treated each document as a sequence of bytes, with no tokenization or parsing of tags or meta-information. This approach was used exclusively for the adhoc web, diversity and relevance feedback tasks, as well as to the batch legal task: the ClueWeb09 and Tobacco collections were processed end-to-end and never indexed. We did the interactive legal task in two phases: first, we used interactive search and judging to find a large and diverse set of training examples; then we used active learning process, similar to what we used for the other tasks, to find find more relevant documents. Finally, we fitted a censored (i.e. truncated) mixed normal distribution to estimate recall and the cutoff to optimize F_1, the principal effectiveness measure.

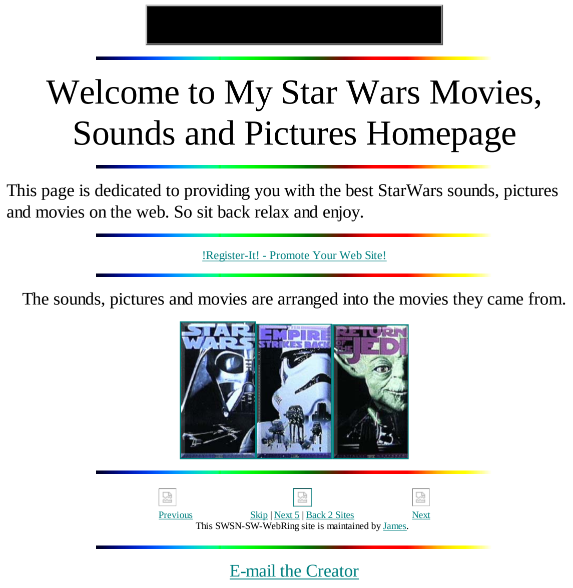
2 Processing ClueWeb09 for Web and Relevance Feedback

We used all the English documents in the full (category A) ClueWeb09 collection. The four distribution drives were mounted on a standard PC with Intel E7400 2.80GHz dual-core processor, 4GB RAM. Decompressing the 12TB of data using gzip requires about 12 hours using both cores; the learning method (for 50 topics in parallel) adds about 6 hours to this time. That is, the score for every document in the collection with respect to every topic is computed in about 18 hours.

To achieve this processing speed, it was necessary to simplify and streamline the learning method. We explain the simplified method along with the more "heavyweight" method from which it was derived in Section 5. To test and tune our approach we first used previous TREC adhoc and web collections. We then composed 67 queries that anticipated the TREC 2009 Web queries as well as we were able, and processed them on the ClueWeb09 corpus (Table 1 on page 1). A cursory examination of the results indicated that, while our learning method was competitive on previous TREC collections, when run on ClueWeb09 it yielded almost entirely spam or low-quality "junk" web

star wars
star wars sdi
star wars luke
sdi
selective disseminatinon
spock
spock kirk
spock benjamin
obama
obama japan
barack obama
capital
capital city
capital assets
money
money pink floyd
money beatles
money spinal tap
spinal tap
spinal tap procedure
spinal tap lyrics
jaguar
jaguar xj
jaguar cat
jaguar fender
fender
fender bender
fender gibson
gates
gates fences
gates steve
windows
windows doors
windows os
apple
apple records
apple computer
macintosh
macintosh apple
apple macintosh
dead poets
vacuum
vacuum cleaner
high vacuum
vacuum
stream
stream process
stream creek
stream education
honda stream
fish
fishing
go fish
fish episodes
whip
whip egg
whip crop
whip topping
party whip
whip it
bull whip
WHIP 1350 AM
whip flagellate
chain whip
The Whip
whip antenna
WHIP walks hits inning pitched

Tab. 1: Pilot Queries composed prior to TREC 2009.



[redacted]

Welcome to My Star Wars Movies, Sounds and Pictures Homepage

This page is dedicated to providing you with the best StarWars sounds, pictures and movies on the web. So sit back relax and enjoy.

!Register-It! - Promote Your Web Site!

The sounds, pictures and movies are arranged into the movies they came from.

Previous Skip | Next 5 | Back 2 Sites Next

This SWSN-SW-WebRing site is maintained by James.

E-mail the Creator

Fig. 1: Low-quality result for "star wars" query.

pages (Figure 1 on page 2).

We anticipated that Wikipedia (which is a subset of the ClueWeb09 collection) would yield higher-quality results, but low recall, missing some topics entirely. For this reason, we did two runs: one fetched the top 10,000 documents for each topic from the entire collection; one fetched the top 10,000 Wikipedia articles for each topic. The Wikipedia results were used for pseudo-relevance feedback, but the collection was not processed again. Instead we used machine learning to re-rank the 20,000 documents per topic, and submitted the top 1000.

For the relevance feedback track, we used much the same method, but used the supplied feedback documents instead of pseudo-relevance feedback. For the Web diversity task, we re-ranked the 20,000 using a naive Bayes classifier designed to exclude duplicates.

3 Processing Tobacco for Batch Legal

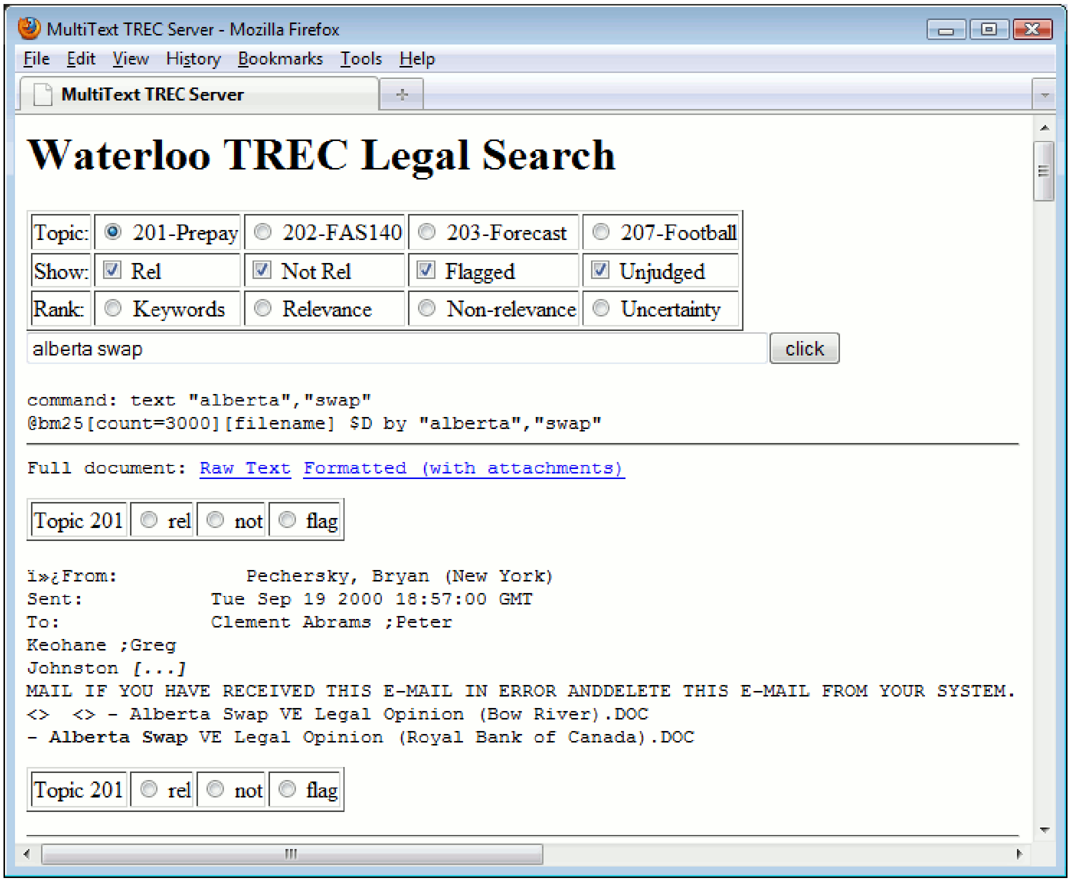
The batch legal task used some 8M documents from the tobacco collection. We ran three spam filters over every document as if it were spam: our on-line logistic regression filter from TREC 2007 [3]; a naive Bayes spam filter modeled after Graham and Robinson [7, 11]; an on-line version of the BM25 relevance feedback method; batch logistic regression, as implemented by the Liblinear package. [1] The results were calibrated to log-odds using 2-fold cross validation and fused using logistic regression [9] and, for a different run, reciprocal rank fusion [4]. The cutoff value of k was chosen so as to optimize F_1 using two-fold cross validation. (In retrospect, we see that this was a mistake, as the training data grossly underestimates the number of relevant documents.) Our third and final run used batch logistic regression trained on the entire training set (as opposed to half, for two-fold cross validation). No calibration was done, and cutoff value k was determined using 2-fold cross validation with the same method. The cutoff value for "highly relevant" was set arbitrarily to the value $0.1k$.

4 Processing Enron for Interactive Legal

The interactive legal task used a new version of the Enron collection with about 800k "documents." During the course of doing the task, it became apparent that about half the documents were incorrectly processed duplicates of others in the collection (which were correctly processed). The upshot is that there are about 250K unique messages, and about 100K unique attachments, many of which are vacuous. We were assigned 4 of the 8 topics: 201 (Prepay transactions), 202 (FAS 140/125 transactions), 203 (Financial forecasts), and 207 (Football).

Our approach consisted of two phases: interactive search and judging, and interactive learning. Our interactive search and judging used essentially the same tools and approach as we used in TREC 6 [6, 13] to prepare an

[1] http://www.csie.ntu.edu.tw/~cjlin/liblinear/


MultiText TREC Server - Mozilla Firefox
File Edit View History Bookmarks Tools Help
MultiText TREC Server
click

Waterloo TREC Legal Search

Topic:	● 201-Prepay	○ 202-FAS140	○ 203-Forecast	○ 207-Football
Show:	☑ Rel	☑ Not Rel	☑ Flagged	☑ Unjudged
Rank:	○ Keywords	○ Relevance	○ Non-relevance	○ Uncertainty

alberta swap

```
command: text "alberta","swap"
@bm25[count=3000][filename] $D by "alberta","swap"
```

Full document: Raw Text Formatted (with attachments)

Topic 201	○ rel	○ not	○ flag

```
ï»¿From:          Pechersky, Bryan (New York)
Sent:           Tue Sep 19 2000 18:57:00 GMT
To:             Clement Abrams ;Peter
Keohane ;Greg
Johnston [...]
MAIL IF YOU HAVE RECEIVED THIS E-MAIL IN ERROR ANDDELETE THIS E-MAIL FROM YOUR SYSTEM.
<>  <> - Alberta Swap VE Legal Opinion (Bow River).DOC
- Alberta Swap VE Legal Opinion (Royal Bank of Canada).DOC
```

Topic 201	○ rel	○ not	○ flag

Fig. 2: Interactive search and judging interface

independent set of qrels for the adhoc task. We used the Wumpus search engine [2] and a custom html interface that showed hits-in-context and radio buttons for adjudication (Figure 2 on page 3). Available for reference were links to the full text of the document and to the full email message containing the document, including attachments in their native format. The resulting qrels were used to train an on-line logistic regression spam filter.

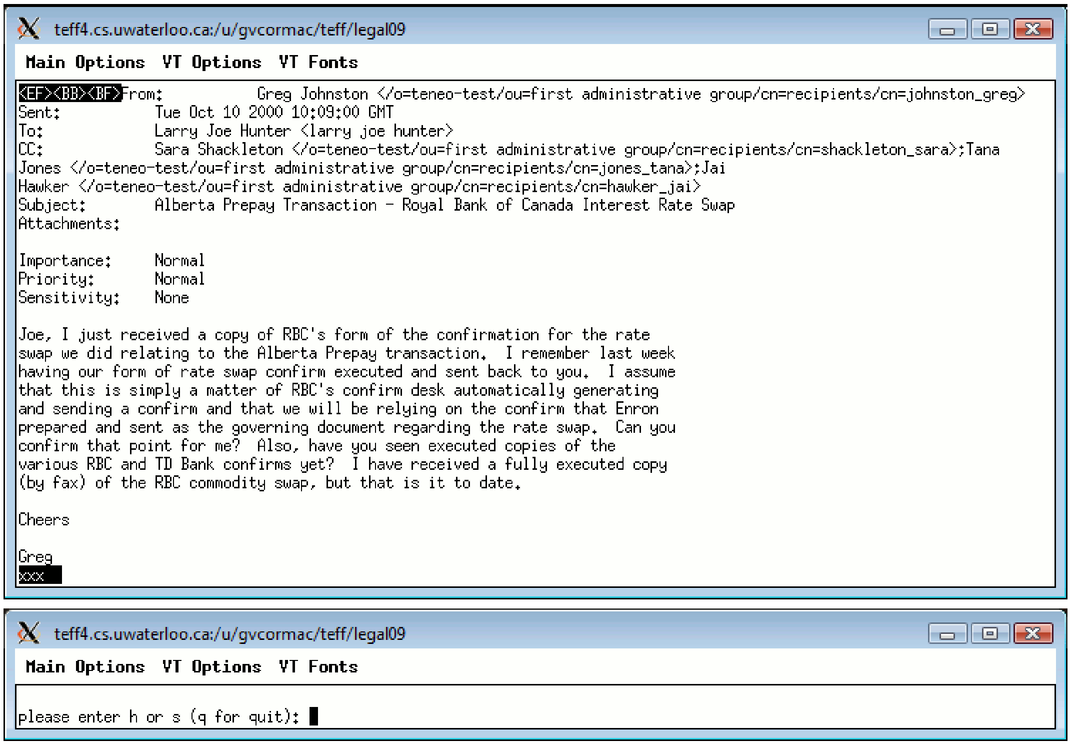
The logistic regression spam filter yields an estimate of the log-odds that each document is relevant. We constructed a very efficient user interface to review documents selected by this relevance score. The primary approach was to examine unjudged documents in decreasing order of score, skipping previously adjudicated documents. Each document was rendered as text and the reviewer hit a single key ("s" for relevant; "h" for not relevant, see Figure 3 on page 4) to adjudicate the document and move on to the next record. About 50,000 documents were reviewed, at an average rate of 20 documents per minute (3 seconds per document). We also examined documents in different orders; in particular, we examined documents with high scores that were marked "not relevant" and documents with low scores that were marked "relevant". From time to time we recomputed the scores by running training the filter on the augmented relevance assessments. From time to time we revisited the interactive search and judging system, to augment or correct the relevance assessments as new information came to light.

Our original intent was to judge the top-ranked documents in each run, to estimate the density of relevant documents as a function of releance score, and to select the appropriate cutoff to optimize F_1. However, when we estimated the density of relevant documents we found that it was feasible to perform manual adjudication well beyond the optimal cutoff. The upshot is that we reviewed every document that we submitted as relevant for each of the topics, and the number of relevant documents we found agrees well with our statistical estimate.

5 Learning Details

The general learning approach (from [1], review copy available on request) was the same for all tasks, with some differences in detail. We describe the approach to feature engineering taken for each task, followed by particular

[2] http://www.wumpus-search.org/


teff4.cs.uwaterloo.ca:/u/gvcormac/teff/legal09
Main Options VT Options VT Fonts

```
<EF><BB><BF>From:          Greg Johnston </o=teneo-test/ou=first administrative group/cn=recipients/cn=johnston_greg>
Sent:           Tue Oct 10 2000 10:09:00 GMT
To:             Larry Joe Hunter <larry joe hunter>
CC:             Sara Shackleton </o=teneo-test/ou=first administrative group/cn=recipients/cn=shackleton_sara>;Tana
Jones </o=teneo-test/ou=first administrative group/cn=recipients/cn=jones_tana>;Jai
Hawker </o=teneo-test/ou=first administrative group/cn=recipients/cn=hawker_jai>
Subject:        Alberta Prepay Transaction - Royal Bank of Canada Interest Rate Swap
Attachments:

Importance:     Normal
Priority:       Normal
Sensitivity:    None

Joe, I just received a copy of RBC's form of the confirmation for the rate
swap we did relating to the Alberta Prepay transaction.  I remember last week
having our form of rate swap confirm executed and sent back to you.  I assume
that this is simply a matter of RBC's confirm desk automatically generating
and sending a confirm and that we will be relying on the confirm that Enron
prepared and sent as the governing document regarding the rate swap.  Can you
confirm that point for me?  Also, have you seen executed copies of the
various RBC and TD Bank confirms yet?  I have received a fully executed copy
(by fax) of the RBC commodity swap, but that is it to date.

Cheers

Greg
xxx
```

teff4.cs.uwaterloo.ca:/u/gvcormac/teff/legal09
Main Options VT Options VT Fonts

```
please enter h or s (q for quit): █
```

Fig. 3: Minimalist review interface

machine learning methods.

5.1 Feature Engineering

Each document was treated as a sequence of bytes without regard to markup or other meta-information. Only the first 30K bytes of each message were used. From previous email and Web spam experiments [2, 3], we had reason to believe that truncating each document would improve both efficiency and effectiveness. We used binary features derived from the occurrence or non-occurrence of particular substrings in the text. For the Legal Track (and some phases of the Web track) these substrings were overlapping byte 4-grams. As there are potentially 2^{32} distinct byte 4-grams, we used hashing to reduce the dimensionality of the feature space to the order of 10^8, facilitating the use of a simple array in place of a dictionary data structure. The collisions that result from this dimensionality reduction have negligible impact on classification performance.

For the primary ClueWeb09 retrieval phase, we processed the query and document text (on the fly) to convert upper to lower case, and to treat all strings of non-alphabetic characters as a single space character. We considered *all* substrings of the query text to be features, and constructed a finite-state machine to recognize all such features in the WARC-format source. From a random sample of 250,000 documents, we selected only those features with a document frequency of less than 0.5. A new finite state machine was constructed to recognize this reduced feature set, and run on the WARC representation of the entire corpus. Based on pilot experiments with previous TREC collections, we departed somewhat from the use of pure binary features: we used an exponential term frequency weight, and also considered the position of the feature relative to the beginning of the document. The same approach was applied both to the entire (English) collection and to the Wikipedia subset.

For the second phase of the Web adhoc and Relevance Feedback tasks, we used binary 4-grams as described above. For the Web diversity task, we used binary "word" features instead of byte 4-grams, and clustered results according to the k most discriminative words, for $k = 1, 3, 5$. Our three diversity runs (one for each value of k) consisted of only the top-ranked document in each category.

5.2 Learning Methods

Logistic regression was used as the principal learning method for all tasks. For the most part we used an online gradient descent approach adapted from spam filtering. In adaptive training mode, this method processes the documents in sequence, and applies one gradient descent optimization step for each message processed. In classification mode, it is a generalized linear classifier that estimates the log-odds of relevance. Recall that each feature is binary, and so a message is represented as a vector X consisting of 10^8 zeroes and ones. The vector is sparse, containing at most 32,000 (and usually nowhere near) 32,000 non-zero entries which can be processed efficiently. The classifier consists of a weight vector β such that

$$\mathit{score} = \beta \cdot X \approx \log \frac{\Pr[\mathit{relevant}]}{\Pr[\mathit{not\,relevant}]}\ . \tag{1}$$

The gradient descent update rule is very simple:

$$\beta \leftarrow \beta + (c - \frac{1}{1+e^{-\mathit{score}}}) \cdot X\ , \tag{2}$$

where $c = 1$ if the document is relevant; otherwise $c = 0$. Note that $\frac{1}{1+e^{-score}} \approx \Pr[\mathit{relevant}]$. Because there are in general far fewer relevant than non-relevant documents, we equalized the number by training on a random relevant document immediately after training each non-relevant document.

For the Web tasks, we constructed synthetic training examples as follows. The query itself was taken to be the sole positive example, and 250,000 randomly selected documents were used as negative examples. The resulting classifier was then applied to the entire corpus. The same approach was used for the Wikipedia subset, with negative training examples selected from only that subset. During classification, a priority queue was used to record the doc-ids of the top-scoring 10,000 documents for each query. A second pass was used to fetch these documents for further processing.

For pseudo-relevance feedback, we selected the top several documents from the Wikipedia run, and used them as positive examples. We did not include the original query as an example. We used randomly selected Wikipedia documents (not necessarily from the run) as negative examples. We trained a naive Bayes classifier on these examples, and applied it to the 20,000 Wikipedia and non-Wikipedia results. The top-scoring 1000 documents per topic were returned. We used naive Bayes instead of logistic regression because of its noise-tolerance properties [12], as we expect that some of our pseudo-relevant examples are in fact non-relevant. Feature selection was done on a document-by-document basis: for each document 60 features were used: the 30 with the largest score, and the 30 with the smallest score. Document-by-document feature selection, suggested by Graham [7], is commonly used in email spam filtering. The naive Bayes classifier is the same generalized linear classifier as generated by logistic regression (equation 1), but with

$$\beta_i = \log \frac{|\{\text{relevant documents with } X_i = 1\}| + \epsilon}{|\{(\text{relevant documents with } X_i = 0\}| + \epsilon} - \log \frac{|\{\text{nonrelevant documents with } X_i = 1\}| + \epsilon}{|\{(\text{nonrelevant documents with } X_i = 0\}| + \epsilon}\ . \tag{3}$$

The set of documents used for pseudo-relevance feedback in the Web adhoc task was determined by classification score: all documents whose score different by less than 1 from that of the top-ranked document were used as positive examples. The difference of 1 is somewhat, but not entirely, arbitrary. It corresponds to an odds ratio of about 1.4, which is in the range of what we commonly think of as a “substantive” difference. Intuitively, if there is not much to choose among the top documents, we use many, but if the top documents are distinctive, we use few.

For the Relevance Feedback track, we use the adjudicated-relevant documents as positive training examples, and add the adjudicated-non-relevant documents to the randomly selected negative examples. It is not appropriate to use only the adjudicated non-relevant examples, as they come from a population that is far from representative. But they are important to tilt the classifier away from other documents like them. As the baseline for the relevance track, we used the two top-ranked documents as positive examples for pseudo-relevance feedback.

For the Legal batch task, we used an ensemble of learning methods. We used the supplied qrels as positive and negative examples, and augmented the negative examples with 20,000 documents randomly selected from the corpus. In addition to the logistic regression and naive Bayes methods mentioned above, we used L_2-regularized (batch) logistic regression, and also an adaptive version of the BM25 and Robertson's relevance feedback method [10] to select 20 feedback terms. In order to combine the results of the four methods it is necessary to calibrate their scores. While our on-line logistic regression method yields a log-odds estimate, it may be overfitted to the training examples. L_2-regularized logistic regression mitigates overfitting by limiting the magnitude of β, with the result that the score ceases to be a log-odds estimate. Naive Bayes is well-known to yield poor estimates, and BM25 makes no estimate at all – its score purports to be useful only for ranking. To calibrate the four sets of scores, we perform 2-fold cross validation. The corpus is randomly partitioned into two equal halves, and each classifier is trained on one half and applied to the other, yielding a score s_d for every document d. s_d is converted to a log-odds

Topic	Reviewed	Est. Rel.	Optimal Cutoff	Assessed Rel.	Official Rel.
201	6145	1919	1897	2154	2454
202	12624	8870	8512	8746	9514
203	4369	3286	2741	2719	1830
207 (except URL)	34446	7325	6914	7365	?
207 (total)	n/a (ad hoc rules for URLs)			23252	26419

Tab. 2: Estimated and submitted number of relevant documents

estimate using the formula [9]:

$$\text{logodds}_d \approx \log \frac{|\{\text{nonrlevant documents with } \mathit{score} \geq s_d\}| + \epsilon}{|\{\text{relevant documents with } \mathit{score} \leq s_d\}| + \epsilon} \ . \tag{4}$$

These estimates are averaged to yield an overall log-odds estimate, which is used to rank the final result for our primary submission. For our secondary submission, we combined the uncalibrated scores from 2-fold cross validation using reciprocal rank fusion [4]. For our tertiary submission, we used L_2-regularized logistic regression, without cross-validation. That is, the training examples consisted of all the qrels, plus a sample of negative examples from the whole collection.

6 Interactive Discovery

We used two interactive tools for the interactive legal task: a search-based system and an active-learning-based system. The search system – outlined above – was used primarily at the outset, to explore the dimensions of the topic and to find as many categories of relevant documents as possible. For example, for topic 201, the task was to find documents related to circular prepay transactions. The search system was used to identify particular transactions, to investigate whether they were or were not circular, and to identify the people and sorts of documents related to the transactions. A simple user interface allowed the searcher to record the relevance of each document, and to select or elide previously-judged ones. The learning system was used to identify more potentially relevant documents, to review those documents, and to record relevance assessments for them.

For the most part, searching was done before learning. But the search system was also used when necessary to investigate (and record) the relevance of documents uncovered by learning. And the learning process was repeated many times, each time using the recorded assessments as training examples and capturing more assessments via the user interface.

The technology of the search system is unexceptional and has been documented elsewhere [5]. The learning component was exactly on-line logistic regression with two-fold cross validation, as used for the batch task. On our standard PC platform, the learning system took about 18 minutes to classify the 800,000 documents in the collection. Since we were working on four topics, it was a simple matter to review one topic while the classifier was running on another.

Our original plan was to use these interactive tools to identify as many relevant documents as possible with reasonable effort, and to augment these documents with top-scoring documents so as to improve recall, thus optimizing the F_1 effectiveness measure. To determine how many such documents to include, it is necessary to estimate precision and recall at all possible cutoff points, and pick the best. To do so, we fitted the maximum likelihood normal distribution to the scores of the relevant documents that we found when judging to a particular cutoff, and used the area under the curve to estimate the number of documents beyond the cutoff. This method agreed remarkably well with the total number of documents we actually found, which gives us some reason to believe it is accurate. Somewhat disappointingly, the result indicated that the optimal strategy was to include *no* unassessed documents, as any improvement in recall would be more than offset by a degradation in precision. The estimated number of relevant documents, optimal submission cutoff, and actual number of examined and judged-relevant documents are shown in table 2. At the time of writing, official estimates for the number of relevant documents were not available.

We state some assumptions underlying the method to estimate recall. First, learning methods are, of course, blind to the definition of relevance, and serve only to identify documents "like the training examples." So when we estimate recall, we are really estimating how many documents "like the training examples" there are. If there happen to be some entirely dissimilar documents that are relevant, they will not be counted. These documents might be in a different language, or perhaps in a different format such as an image or multimedia. We may only assume that we have done due diligence in our initial search, so we have reason to believe that an insubstantial

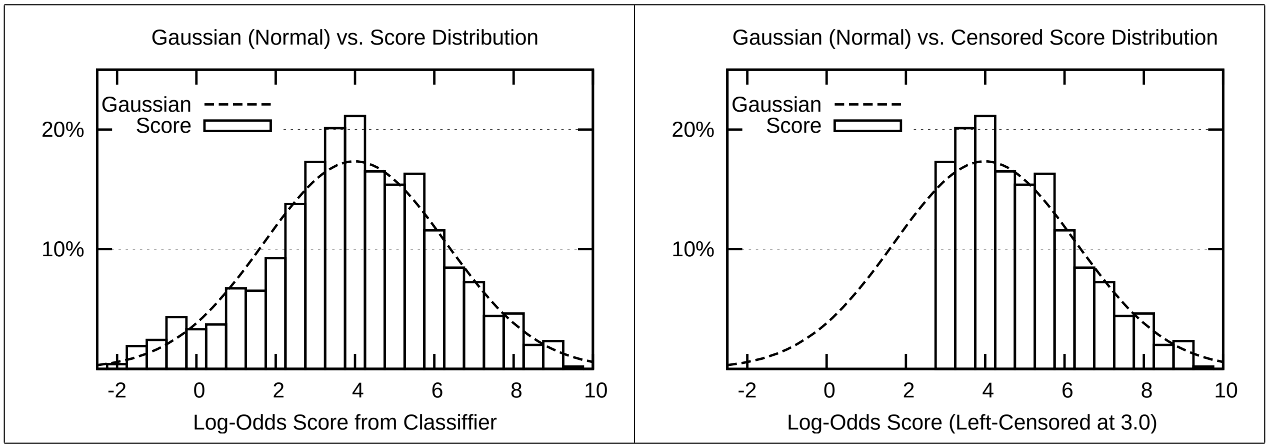


Fig. 4: Fitting a Gaussian distribution. The left panel shows the scores of all judged-relevant documents for topic 201 vs. the maximum likelihood Gaussian distribution. Typically, it is feasible to judge only the top-scoring documents, resulting in a left censored distribution, shown in the right panel. The maximum likelihood fit use used to estimate the number of relevant unjudged documents.

Run	P@10	Run	P@10	Run	P@10
uvamrftop	0.4100	**watprf**	**0.3360**	WatSdmrm3we	0.1640
MS2	0.4060	muadimp	0.3006	UMHOObm25IF	0.1640
yhooumd09BGM	0.4040	Sab9wtBf1	0.2880	pkuSewmTp	0.1480
MSRAC	0.4000	muadibm5	0.2788	pkuStruct	0.1460
yhooumd09BGC	0.3840	Sab9wtBf2	0.2620	UMHOObm25GS	0.1420
watrrfw	**0.3760**	twJ48rsU	0.2380	WatSdmrm3	0.1180
THUIR09An	0.3740	Sab9wtBase	0.2260	UMHOOqlGS	0.1180
MSRANORM	0.3700	THUIR09LuTA	0.2100	pkuLink	0.1160
THUIR09TxAn	0.3640	twCSrs9N	0.2080	uvaee	0.1100
MSRAAF	0.3540	twCSrsR	0.1800	UMHOOqlIF	0.1080
MS1	0.3540	uogTrdphP	0.1680	uvamrf	0.0940
muadanchor	0.3519	yhooumd09BFM	0.1640	WatSql	0.0840
watwp	**0.3516**				

Tab. 3: Web Track Ad hoc results. P@10 results for all Category A ad hoc submissions are reproduced here. **watwp** is our Wikipedia-only run; **watprf** is our pseudo-relevance feedback run, using the top watwp results as seeds; **watrrfw** is the combination of the two using reciprocal rank fusion.

number of dissimilar documents exist. This assumption is not unique to our efforts; it is tacit in the TREC pooling method, and also methods based on sampling like those used to evaluate the Legal track results.Second, we assume that the distribution of log-odds scores for relevant documents is normal. This assumption is generally true for natural phenomena, and appears to hold for the scores from our classifier as well. Figure 4 illustrates the process of fitting a Gaussian to the top-scoring documents that are judged relevant.

Estimating precision is more problematic. Based on our previous experience, we thought it is unlikely that the precision of our human assessment is greater than 0.7, notwithstanding our use of the topic authorities. For topic 203 in particular, we did not think we were able to acquire a firm enough grasp on the notion of a “responsive” document to predict precision greater than 0.5. Statistical estimation is of little help, as we judged every document that was submitted as relevant. The end result hinges on our agreement with the official adjudication.

7 Results

Table 3 shows precision at cutoff 10 for all Category A ad hoc runs, reproduced from the appendices. Our runs, prefixed by **wat** (not **Wat**), show that the method is effective. Our diversity results were unremarkable.

Table 4 shows precision at cutoff 10 for all Category A relevance feedback runs. While the design of the task makes pairwise comparisons difficult, it appears our feedback method is competitive with that of Sabir, and superior

Run	P@10	Run	P@10	Run	P@10
WAT2.WatS.2	**0.5900**	CMIC.CMIC.2	0.3900	WatS.UCSC.1	0.3000
Sab9RF.udel.1	0.5340	CMIC.ilps.1	0.3880	WatS.twen.2	0.2960
Sab9RF.CMIC.2	0.5280	CMIC.ugTr.2	0.3860	WatS.SIEL.1	0.2680
WAT2.UCSC.1	**0.5120**	**WAT2.UPD.1**	**0.3800**	**WAT2.YUIR.1**	**0.2540**
Sab9RF.WatS.2	0.5100	CMIC.UMas.2	0.3580	WatS.twen.1	0.2520
WAT2.udel.1	**0.5040**	CMIC.MSRC.1	0.3580	WatS.fub.1	0.2420
Sab9RF.Sab.1	0.4880	WatS.WatS.2	0.3520	IlpsRF.ilps.2	0.2380
Sab9RF.UCSC.2	0.4800	Sab9RF.YUIR.1	0.3480	IlpsRF.ilps.1	0.2340
WAT2.CMIC.2	**0.4560**	WatS.WatS.1	0.3400	IlpsRF.twen.1	0.2220
CMIC.CMIC.1	0.4420	**WAT2.MSRC.2**	**0.3300**	IlpsRF.twen.2	0.2160
MSRC.CMU.1	0.4360	IlpsRF.WatS.1	0.3280	IlpsRF.fub.1	0.1800
Sab9RF.hit2.2	0.4180	CMIC.udel.2	0.3240	IlpsRF.QUT.1	0.1360
Sab9RF.MSRC.2	0.4080	**WAT2.hit2.2**	**0.3100**	IlpsRF.Sab.1	0.1020
CMIC.udel.1	0.3960	WatS.Sab.1	0.3020		

Tab. 4: Relevance Feedback results. P@10 results for all Category A phase 2 submissions are reproduced here. Our runs have the prefix **WAT2**.

	Predicted (doc.)			Actual (doc.)			Actual (msg.)		
Topic	Recall	Prec.	F1	Recall	Prec.	F1	Recall	Prec.	F1
201	0.9	0.7	0.787	0.843	0.911	0.876	0.778	0.912	0.840
202	0.9	0.7	0.787	0.844	0.903	0.872	0.673	0.884	0.764
203	0.5	0.3	0.375	0.860	0.610	0.714	0.865	0.692	0.769
207	0.95	0.9	0.924	0.896	0.967	0.930	0.761	0.907	0.828

Tab. 5: Legal Interactive

to the rest.

Table 5 shows recall, precision and F_1 for our four Legal Track interactive efforts. The first three columns show our predictions for these measures at the document level. The centre three columns show the the document-level measures computed from the official adjudicated results. The right three columns show message-level measurements. With the exception of topic 203, our recall predictions were consistently optimistic, while our precision predictions were pessimistic. Pessimism outweighed optimism with the net effect that our F_1 predictions were all pessimistic but, with the exception of 203, reasonably accurate. Our submission for 203 was best characterized as a “Hail Mary” play, one that appears to have been successful.

Our Legal Track batch efforts achieved the best overall $F_1@k$ for both relevant and highly relevant documents. However, we argue that the actual numbers – 0.214 and 0.190 respectively – are essentially meaningless, due to assessment error. The reader is referred to the track overview [8] for further details, and a comparison of our batch results with those of the 2008 interactive task.

8 Discussion

One of the questions raised with regard to the ClueWeb09 collection was: how difficult will it be to index the collection? We circumvented this question by applying a sequential content-based classifier to the entire corpus. For 50 or even 500 topics, we suggest this approach is far cheaper than indexing. Furthermore, sequential processing facilitates exhaustive searching of large archives, as occasioned by legal discovery and information archaeology tasks. The methods we have developed are essentially domain independent. The finite-state classifier we used for ClueWeb09 could apply to any corpus; the methods of interactive search and judging and active learning require a certain amount of domain knowledge. We advocate further research into measuring and minimizing the amount of interaction necessary to acquire the necessary domain-specific information.

References

[1] Stefan Buettcher, Clarles L. A. Clarke, and Gordon V. Cormack. *Information Retrieval: Implementing and Evaluating Search Engines.* MIT Press, 2010.

[2] Gordon V. Cormack. Content-based Web spam detection, 2007.

[3] Gordon V. Cormack. University of waterloo participation in the trec 2007 spam track. In *Sixteenth Text REtrieval Conference (TREC-2007)*, Gaithersburg, MD, 2007. NIST.

[4] Gordon V. Cormack, Charles L. A. Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms Condorcet and individual rank learning methods. In *SIGIR '09: Proceedings of the 32nd annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, 2009. ACM Press.

[5] Gordon V. Cormack, Christopher R. Palmer, and Charles L. A. Clarke. Efficient construction of large test collections. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 282–289, 1998.

[6] Gordon V. Cormack, Christopher R. Palmer, and Charles L. A. Clarke. Efficient construction of large tst collections. In *SIGIR Conference 1998*, Melbourne, Australia, 1998.

[7] Paul Graham. Better bayesian filtering. http://www.paulgraham.com/better.html, 2004.

[8] Bruce Hedin, Stephen Tomlinson, Jason R. Baron, and Douglas W. Oard. Overview of the TREC-2009 Legal Track. In *Proceedings of the Eighteenth Text REtrieval Conference*, Gaithersburg, Maryland, 2009.

[9] Thomas R. Lynam and Gordon V. Cormack. On-line spam filter fusion. In *29th ACM SIGIR Conference on Research and Development on Information Retrieval*, Seattle, 2006.

[10] S. E. Robertson. On term selection for query expansion. *Journal of Documentation*, 46:359–364, December 1990.

[11] G. Robinson. A statistical approach to the spam problem. *Linux Journal*, 107:3, March 2003.

[12] D Sculley and Gordon V. Cormack. Filtering spam in the presence of noisy user feedback. *Tufts University*, 2008.

[13] E.M. Voorhees. Variations in relevance judgments and the measurement of retrieval effectiveness. *Information Processing and Management*, 36(5):697–716, 2000.