Figure 1: M.S. Thesis Approval Form
Master Thesis:
Applying EM to Compute Document Relevance from
Crowdsourced Pair Preferences

Jie Wu
College of Computer and Information Science
Northeastern University
Boston MA 02115
wu.jie1@husky.neu.edu
Thesis Advisor: Prof. Javed A. Aslam

April 30, 2013

Abstract

Traditional Information Retrieval (IR) systems evaluate over absolute relevance judgments or multi-grade judgments by hiring expert assessors. More recently there has been considerable interest in inferring document relevance, for which no true relevance labels are available, based on noisy crowdsourcing data. We introduce an algorithm that, using Expectation Maximization (EM) and the Elo Rating System on pairwise preference judgments, acquires document relevance from crowdsourced data only. We collect pairwise ternary judgments from workers with unknown accuracy, calculate the pairwise probabilities of each document pair, and transform them into document relevance probabilities. Each judgment is presented in the form, given query q, document a is more relevant to q than document b, or document b is more relevant than document a, otherwise they tie. We hypothesize that it is easier for the crowd workers to make preference judgments than absolute judgments, and preference judgments may offer more accurate relevance assessment. We use EM to calculate the accuracy of the crowd workers, estimate document preference probabilities, and to provide input to the Elo Rating System which produces a ranked document list.

1 Introduction

Information retrieval test collections traditionally contain a document corpus, a set of queries, and a group of absolute binary judgments or multi-grade judgments provided by experts in the query topics. With
the large size of recent data corpora and the number of queries, gathering labels from experts in each query topic is too expensive, but crowd sourcing may provide solutions to this problem. Queries are broadcast to a large unknown group of workers in the form of an open call for solutions. Since the background of the workers is unknown, we assume they are not the experts in the field of the query topic, and so payments are small. There are many workers who would like to devote their knowledge and time, and so we can get as much data as we need. TREC 11 crowd overview defines crowd sourcing as “a collection of mechanisms and associated methodologies for scaling and directing crowd activities to achieve some goal(s).”

Recently, there has been considerable interest in utilizing preference judgments [11] for applications in Information Retrieval. Ben Carterette suggests that preference judgments have the following advantages:

- “Assessors make preference judgments faster than absolute judgments on a graded scale”
- “When preferences are transitive, they can be mapped to a measure of individual document utility that can be understood as an absolute relevance judgment”
- “Preferences more naturally reflect objective functions in pairwise learning-to-rank algorithms”

EM is normally used to solve problems which include multiple unknown parameters and some known observations. In our case the known observations are pair preferences by specific workers, and the unknown parameters are worker quality and the probability of each preference class for each document pair. EM obtains accurate preference probability even in the presence of low quality crowd workers. However, in order to compare our results with the gold answer sheet qrel file, we need to produce a ranked list.

The Elo Rating System (Elo) is a method for calculating the relative skill levels of players in two-player games such as chess. It uses numerical number to represent players’ rating, and updates players’ rating after each game or tournament. It was adopted by the World Chess Federation in 1970. The foundational ideas within Elo have also been adopted by many different organizations and different rating systems. In our system we use Elo produce a ranked document list based on preference judgements or preference class probabilities.

We propose that a system which combines EM with Elo will outperform plain Elo in the presence of noisy preferences. Elo regards all input preference judgements equally, but EM accounts for worker quality and ignores or flips the judgments of low quality workers. A ranked list produced by just Elo should be less accurate than one produced by Elo run on the output of EM.

In the next section we discuss recent research which is also relevant to our topic.

Section 3 explains the experimental setup, including data collection, and our EM-based algorithm.

Section 4 shows the Elo Rating System and the use of it to transform pairwise probabilities to a ranked list of documents.

Section 5 collects results of Elo Rating System and the results of using EM plus Elo. The evaluation is also discussed in this section. We used trec_eval to evaluate both systems by calculating the Mean Average
2 Related Work

Gabrilla Kazai suggests a way of using the EM algorithm to create relevance judgments for IR test collections [15]. She applies the Majority Voting (MV) and EM methods to crowd sourced labels and compares the derived relevance judgments with a gold standard set. Her empirical results show that EM offers more reliable relevance estimations than MV, especially when the labels collected for a document are few or varied in quality. When worker average expertise is close to random, 0.5, and two judgments are gathered for each document, both MV and EM exhibit poor performance. If the number of labels per document is kept small, but worker average expertise is increased to 0.6 or higher, EM outperforms MV. If the number of labels per document is increased to 10, and the workers average expertise is 1, both methods obtain perfect accuracy.

Dai and Pavlu expanded the EM algorithm to infer score distributions [14]. IR applications return a document ranked list to the user as a response to the query. Ideally this ranked list contains only relevant documents; however, in reality both relevant documents and nonrelevant documents are present. Modeling and inferring the distribution of relevant and nonrelevant documents over scores with a reasonable precision can be highly beneficial. They imposed the constraint that a document should have the same probability of a relevance when it is returned by different systems. Their extended EM outperforms the regular EM in terms of inferring precision-recall curves and estimating expected average precision.

Despite the success of research involving EM, it has been noted that EM suffers from treating all data equally, being very sensitive to initialization, and converging to a local optimum instead of the global one [20, 3, 4, 5].

Crowdsourcing is gradually becoming more common as a method to gather data for the evaluation of search engines, however, questions are raised by the lack of the knowledge of the backgrounds of the crowd workers. [16] investigates factors which may affect label quality. They did controlled crowdsourcing experiments from three angles: payment, worker quality, and the required effort to make a judgment. Their result shows that higher pay encourages better work, especially of the higher quality workers, but it also attracts more unethical workers. Required effort also matters, and better results can be produced when workers are not overwhelmed.

[17] investigates an effectively IR system in the context of book search evaluation using crowdsourcing. One of the big challenge of book search systems is the sheer effort of reviewing whole books and rendering relevance judgments for pages across a large number of retrieved books. The paper concludes the reliability of crowdsourced labels vary and using HITs to control the engagement of crowd workers and the quality of their output is necessary. Their analysis also highlights the varying resulting labels and a danger of relying upon a single metric in system evaluation to draw conclusion.
Researchers have also investigated methods to evaluate search engines using preference judgments. Though at first, their use seems it would polynomially increase the number of required judgments, [12] shows that by using inferences and clever selection of which pairs to judge, it is not necessary to compare all pairs of documents in order to apply evaluation methods. There are also several learning algorithms based on preferences, such as the ranking SVM [8] and RankNet [7]. [13] employs PageRank to minimize the number of judgments required to evaluate systems.

Dongqing Zhu and Ben Carterette did an analysis to explore the assessor behavior in crowdsourced preference judgments. They explore how to make the research results layout more user-friendly by varying the position of images relative to ranked results [21].

Panagiotis Ipeirotis showed that even repeated labeling does not always get better quality labels than single labeling, but it can be preferable when labels are noisy. Repeated labeling can show its considerable advantage when it is used on a carefully chosen data set. [18]

[19] Adam Tauman Kalai introduces an algorithm that given n objects, learns a similarity matrix over all \( n \times 2 \) pairs using crowdsourcing data only. The algorithm samples responses to adaptively chosen triplet-based queries, which are in the form “is object a more similar to b or to c?” The result shows their output, an embedding of the objects into Euclidean space, captures prominent and subtle features across a number of domains.

[1] propose a novel method to evaluate rank accuracy for recommender systems, *expected discounted rank correlation*, which handles incomplete pairwise preferences in ground truth. In reality, many applications only obtain implicit user feedback, which does not provide enough comprehensive information to generate a ordered list among documents or items. They collect ground truth preference using the items that user visited or purchased.

Learning To Rank (LTR) is also used to generate ranked list using preferences. [2] suggests an efficient approach to deal with the traditional LTR problems:

1. A preference function may not induce a linear ordering.
2. Finding a linear ordering with as few pairwise misrankings as possible with respect to a preference function is NP-complete.

## 3 Experimental Setup

### 3.1 Data Collection

The queries and document corpora used in our experiments come from the TREC 12 Crowdsourcing Track. TREC is an annual Information Retrieval (IR) competition and conference to support the IR research community’s effort to improve text retrieval methodologies for large text collections. TREC provides vetted queries and corresponding document corpora with expert relevance judgments as a benchmark for
the evaluation of such methodologies. TREC is co-sponsored by the National Institute of Standards and Technology (NIST) and U.S. Department of Commerce.

- **Topic:** Salvaging, shipwreck, treasure
  - **Description:** Find information on shipwreck salvaging: the recovery or attempted recovery of treasure from sunken ships.
  - **Narrative:** A relevant document will provide information on the actual locating and recovery of treasure; on the technology which makes possible the discovery, location and investigation of wreckages which contain or are suspected of containing treasure; or on the disposition of the recovered treasure.

- **Topic:** Three Gorges Project
  - **Description:** What is the status of The Three Gorges Project?
  - **Narrative:** A relevant document will provide the projected date of completion of the project, its estimated total cost, or the estimated electrical output of the finished project. Discussions of the social, political, or ecological impact of the project are not relevant.

- **Topic:** Creativity
  - **Description:** Find ways of measuring creativity.
  - **Narrative:** Relevant items include definitions of creativity, descriptions of characteristics associated with creativity, and factors linked to creativity.

- **Topic:** Carbon monoxide poisoning
  - **Description:** How widespread is carbon monoxide poisoning on a global scale?
  - **Narrative:** Relevant documents will contain data on what carbon monoxide poisoning is, symptoms, causes, and/or prevention. Advertisements for carbon monoxide protection products or services are not relevant. Discussions of auto emissions and air pollution are not relevant even though they can contain carbon monoxide.

- **Topic:** UV damage, eyes
  - **Description:** Find documents that discuss the damage ultraviolet (UV) light from the sun can do to eyes.
  - **Narrative:** A relevant document will discuss diseases that result from exposure of the eyes to UV light, treatments for the damage, and/or education programs that help prevent damage. Documents discussing treatment methods for cataracts and ocular melanoma are relevant even when a specific cause is not mentioned. However, documents that discuss radiation damage from nuclear sources or lasers are not relevant.
• **Topic:** Profiling, motorists, police
  
  – **Description:** Do police departments use “profiling” to stop motorists?
  
  – **Narrative:** A relevant document will report or discuss police department criteria for identifying motorists considered likely to be carrying contraband. Documents discussing the detention of individuals by foreign security forces are not relevant.

• **Topic:** Tourism, increase
  
  – **Description:** What countries are experiencing an increase in tourism?
  
  – **Narrative:** A relevant document will name a country that has experienced an increase in tourism. The increase must represent the nation as a whole and tourism in general, not be restricted to only certain regions of the country or to some specific type of tourism (e.g., adventure travel). Documents discussing only projected increases are not relevant.

• **Topic:** Women clergy
  
  – **Description:** What other countries besides the United States are considering or have approved women as clergy persons?
  
  – **Narrative:** To be relevant, a document must indicate either a country where a woman has been installed as clergy or a country that is considering such an installation. The clergy position must be as church pastor rather than some other church capacity (e.g., nun or choir member).

• **Topic:** Tourists, violence
  
  – **Description:** Where are tourists likely to be subjected to acts of violence causing bodily harm or death?
  
  – **Narrative:** A relevant document must contain accounts of known harm to tourists. Evidence of single, isolated incidents are not relevant.

• **Topic:** Stirling engine
  
  – **Description:** What new developments and applications are there for the Stirling engine?
  
  – **Narrative:** Any discussion of new developments and applications of the Stirling engine (also known as the Stirling cycle) are relevant.

The above list shows the 10 queries, descriptions, and narratives. The queries involve different knowledge fields. Each query has a title, a description, and a short narrative describing the exact standard for categorizing documents as relevant or irrelevant. We provide TREC’s description and narrative to our crowdsourcing labelers to help them understand the query. Relevant documents must strictly fit the narrative of the query.

There are approximately 1100 to 3000 documents for each query. For each document we choose five other documents to pair it with, and for each pair we collect at least four judgments from different labelers.
We collect data through Amazon Mechanical Turk, which is an online crowdsourcing marketplace for harvesting human intelligence. In many areas, human intelligence outperforms computing methods. Amazon Mechanical Turk provides enough quickly-available low-cost on-demand workers for our needs, and a programmable API to build Human Intelligence Tasks (HIT).

In order to get pair preference judgments, we ask “given query x, do you think document A is more relevant than B or document B is more relevant than A, or tie? ”

Figure 2 shows the instructions provided to our crowd workers. We ask workers to select the document that is more related to the query topic. For example, there might be a document that provides precise and detailed information on an unrelated topic, but workers are asked to prefer a short and badly written document if it is relevant to the query.

Figure 2 is an example interface for the HIT we posted on Amazon Mechanical Turk. It has topic query and a description at the top of page. The two documents are shown side by side, and four radio buttons are displayed above the documents. The leftmost and rightmost buttons are labeled “This one” (referring to the document on that side of the page), and the middle buttons are labeled “They are equally good” and “They are equally bad”. If the worker prefers one document over the other, then he should click the corresponding “This one” button. If the worker cannot decide between the documents, then he should select one of the
middle buttons, indicating a tie of relevance or nonrelevance.

3.2 Trap questions

Since the quality of crowdsourcing work is not guaranteed, Amazon Mechanical Turk allows researchers or companies to only pay for quality work by accepting or rejecting the received work after reviewing. In our project, in order to filter out low quality crowd workers, we set up trap questions for each assignment. Each HIT is composed of 15 questions and an additional 5 trap questions. Workers were paid $0.15 for each completed HIT. If a worker took more than thirty minutes to complete all 20 questions or failed to answer a fixed threshold of trap questions correctly, then the answers given for the entire hit were rejected.

Trap questions are created before posting any HITs on Amazon Mechanical Turk. Some expert judges are hired to assess the ground truth for preferences of these document pairs. We used 5 IR graduate students as the experts for creating the trap questions. Each HIT includes 5 trap questions mixed in so that workers do not know which questions are trap questions, and they do not judge trap questions more carefully than data collection questions. We are able to assess the quality of the preference judgements submitted for each HIT by including trap questions in every HIT.

We show the statistic numbers in Table 1 showing the acceptance percentage. It shows the total number of collected judgments and the number that accepted. The last column show the accepting rate. Since the diffi-
Table 1: Data Collection & Acceptance Rate

certainty of each query and trap questions are vary query by query, we got acceptance rate from 55.4% to 78.98%.

3.3 Pair Picking

As we discussed before, the major downside of using pair preferences is the scale of the data set. In order to optimize information gain, relative to the number of pair preference judgments, we tried different pair-picking strategies. We tested our pair-picking strategies on one of the queries from TREC 11, “Home Remedies For Canker Sores”. We gathered pair preference judgments for all the potential pairs, ran the preferences through Elo, and evaluated the resulting ranked list Mean Average Precision (MAP). Elo, as we discussed in the introduction section, is a method using numerical rating to evaluate player’s relative skill level. Player’s rating will be updated after a game or a tournament in which this player was involved. MAP is a standard measure to evaluate IR system. It uses a 0 to 1 value to represent IR system. The higher this number is the better. Details about MAP will be discussed in the evaluation section. After comparing the results from different pair-picking strategies using MAP, we chose a random-intelligent way of picking pairs.

Query topic: Home Remedies For Canker Sores
Description: What are some home remedies for canker sores?
Narrative: What are canker sores? Are they painful? Where do they develop on the body? What non-medications could be used to treat them? What medications are used for treatment? Are canker sores contagious? Can canker sores heal by them self?
Total number of documents: 105.
Number of relevant documents: 33.

We used Mechanical Turk to collect preferences for all 5,460 potential pairs of the 105 documents provided for this query. Each pair has at least 4 judgments given by different crowd workers.

A potential problem with Elo arises when preferences do not produce a well-connected graph among judged items. Elo may categorize items into cohorts, and generate an accurate ranked list within each while
inaccurately ordering the cohorts themselves.

Consider a chess competition including 6 international experts and 6 undergraduate students. The experts play matches amongst themselves, while the students do the same. With the resulting player preferences, Elo will generate a precise ranked list of either the experts or the students. Without games between experts and students, Elo will not be able to produce a precise ranking which includes all the players. It is possible that the highest ranked student will appear at the top of the list of all players because he won many games, even if he only played against lower level players. We suggest that if each player always competes with those players that have higher ranking, the resulting pairs would help Elo to find the correct overall relationships among different levels of players.

We hypothesize that we can use a linear number of pairs instead of polynomial to get reasonable results using our system.

We tested three pair picking strategies on the 18,486 pair preferences gathered for this purpose. The first strategy simply chooses all potential pairs. As long as the input preferences are not overly noisy, this should produce a ranked list with the theoretical-best MAP because the graph among documents is complete. In the second strategy (Random Strategy), for each document we choose five other documents to make pairs with. Hence the pair scale is $5 \times n$ instead of $n \times 2^2$. We directly address problem explained above in our third strategy (Random Intelligent Strategy) by forcing each document to only be paired with higher ranking documents.

**Intelligent-Random pair picking strategy** In order to pair documents with higher ranked documents only, we initialize a ranked list with BM25 scores. BM25 is a popular ranking function used by IR system to rank matching documents according to their relevance to a given query. It helps us generate a accurate document relevance ranked list. The MAP value for this ranked list is 0.25, as calculated by trec_eval. For the top 6 documents, we make all possible pairs. For each other document $d$, we randomly choose 5 documents with higher BM25 relevance than $d$. This method relies on the assumption that BM25 scores represent some approximation of ground truth relevance for a given document.

<table>
<thead>
<tr>
<th></th>
<th>All Data</th>
<th>Random</th>
<th>Random &amp; Intelligent</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAP</td>
<td>0.8455</td>
<td>0.5370</td>
<td>0.8047</td>
</tr>
<tr>
<td>Retrieved</td>
<td>105</td>
<td>105</td>
<td>105</td>
</tr>
<tr>
<td>Relevant</td>
<td>33</td>
<td>33</td>
<td>33</td>
</tr>
<tr>
<td>Retrieved-relevant</td>
<td>33</td>
<td>33</td>
<td>33</td>
</tr>
<tr>
<td>Precision at 5 docs</td>
<td>1.0</td>
<td>0.80</td>
<td>1.0</td>
</tr>
<tr>
<td>Precision at 10 docs</td>
<td>0.90</td>
<td>0.60</td>
<td>1.0</td>
</tr>
<tr>
<td>Precision at 15 docs</td>
<td>0.867</td>
<td>0.467</td>
<td>0.93</td>
</tr>
<tr>
<td>Precision at 20 docs</td>
<td>0.90</td>
<td>0.35</td>
<td>0.80</td>
</tr>
<tr>
<td>Precision at 30 docs</td>
<td>0.867</td>
<td>0.467</td>
<td>0.733</td>
</tr>
</tbody>
</table>

Table 2: Data Collection & Acceptance Rate

We ran Elo on the three sets of pairs produced by our strategies, and evaluated each by calculating their MAP value with trec_eval. Table 2 shows the results of Elo on the three sets of pairs. Running Elo
on all the potential pairs got the highest MAP value, which matched our assumptions. The gap between 0.8455 and 1.0 is caused by noise in the crowdsourcing judgments. The MAP value produced by the Random Strategy was dramatically lower, 0.537. The Random-Intelligent strategy produced a MAP competitive with running on all data, which means that the pairs we picked are able to provide enough information to infer document ranking among different relevance cohorts. In all following experiments, we collected data using the Random-Intelligent pair picking strategy to minimize the required number of judgements.

Table 3 lists the sizes of the data sets we collected for the ten queries. The “Target” column is the total number of preferences, including trap questions, that were collected.

\[
\text{Target} = \frac{\text{Docs} \times 5 \times \text{pairs} \times 4 \times \text{judgments}}{(15/20)}
\]

<table>
<thead>
<tr>
<th>Topic</th>
<th>Num of Docs</th>
<th>Target</th>
<th>Num of Judgments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Salvaging, shipwreck, treasure</td>
<td>2056</td>
<td>54826</td>
<td></td>
</tr>
<tr>
<td>Three Gorges Project</td>
<td>1235</td>
<td>32933</td>
<td></td>
</tr>
<tr>
<td>Creativity</td>
<td>2992</td>
<td>79786</td>
<td></td>
</tr>
<tr>
<td>Carbon monoxide poisoning</td>
<td>1136</td>
<td>30293</td>
<td></td>
</tr>
<tr>
<td>UV damage, eyes</td>
<td>1528</td>
<td>40746</td>
<td></td>
</tr>
<tr>
<td>Profiling, motorists, police</td>
<td>2503</td>
<td>66746</td>
<td></td>
</tr>
<tr>
<td>Tourism, increase</td>
<td>1798</td>
<td>47946</td>
<td></td>
</tr>
<tr>
<td>Women clergy</td>
<td>1404</td>
<td>37440</td>
<td></td>
</tr>
<tr>
<td>Tourists, violence</td>
<td>2020</td>
<td>53866</td>
<td></td>
</tr>
<tr>
<td>Stirling engine</td>
<td>1588</td>
<td>42346</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Data collection

3.4 Phase I: EM-based algorithm

Expectation Maximization (EM) is an iterative method that attempts to find the maximum likelihood estimator of a parameter \( \theta \) of a parametric probability distribution. It has two steps: expectation (E) uses estimated variables to evaluate the expectation of the likelihood of other variables, and maximization (M) maximizes the expected likelihood generated in the E step.

In our EM model, the observed judgments from Mechanical Turkers are the given input. Document pair preference probability and worker quality are latent variables.

Since document pair preference judgments are the only known information, we expect all the document pair preference probabilities to be the same. We order documents of each document pair in alphabetic order, which means the relevance relationship between the two documents in each pair is totally unpredictable. Then for any arbitrary document pair \( i \), the probability that the first document is more relevant than the second document is the same as it is less relevant than the second document. With initial guesses for the pair preference probabilities, the expectation values of each crowd worker’s confusion matrix is calculated.
by using the observations. Worker quality confusion matrix is calculated based on how much worker agrees with the system believes, which is the pair preference probabilities.

With the expectation confusion matrix of each worker in hand, the maximum likelihood of document pair preference probabilities are estimated using the expectation values. We consider the new worker quality as the ground truth and re-estimate the maximum likelihood of document pair preference probability by agreeing with the majority of the high quality workers’ answers. We value the high quality workers’ answer and lower the weights of the low quality workers’ answers.

Likely, treating the new pair preference probability as the ground truth, the maximum likelihood of worker quality can be re-calculated. The estimation of document pair preference probability and the expectation of worker quality will be repeated until the result converged.

We take a similar approach as in [15] to implement our EM algorithm, but document pairs are the objects of interest instead of the documents themselves.

Assuming there is a set of $N$ document pairs and $M$ workers. $G$ is the set of preference classes. We assign each worker $j$ a $G \times G$ latent confusion matrix. Each element $c_{l,g}^j$ is the probability that worker $j$ labels the document as $l$, given ground truth $g$.

$pr(R_i = g)$ refers to the probability that document pair $i$ has the true preference class $g$.

$p_g$ is the prior probability distribution of preference class $g \in G$.

$n_{ij}^l$ is a binary digit indicating whether worker $j$ has labeled document pair $i$ as $l$, where 1 indicates “yes”.

---

Figure 4: The EM algorithm.
3.4.1 Algorithm Detail

There are four steps in the EM algorithm. Our first experiment assumes there are only preference judgments, and does not consider the “Tie” situation. Hence, for each arbitrary pair, there are two preference classes in set \( G \), “more relevant” or “less relevant”.

\[ G = \{mr, lr\} \]

Step 1: Initialization. We initialize the preference class for each document pair. For all document pairs \((doc_1, doc_2)\), we assume the probability that \(doc_1\) is more relevant than \(doc_2\) to be 0.5. Likewise we assume the probability that \(doc_1\) is less relevant than \(doc_2\) to be 0.5.

\[ Pr(R_i = mr) = 0.5 \]
\[ Pr(R_i = lr) = 0.5 \]

With the above initialization we can calculate the \( p_g \) for any \( g \in G \). We calculate this using the sum of all document pair probabilities having class \( g \), divided by the sum of all document pair probabilities for all classes. This denominator equals the number of document pairs, because the sum of the probability distribution for each is 1 (\( \sum_{k \in G} Pr(R_i = k) = 1 \)).

\[ p_g = \frac{\sum_i^N Pr(R_i = g)}{\sum_i^N \sum_{k \in G} Pr(R_i = k)} \tag{1} \]

Step 2: Confusion matrix. Estimate the maximum likelihood of worker quality, represented by a 2 x 2 matrix of \( c_{l,g} \).

\[ M = \begin{bmatrix} c_{mr,mr}^j & c_{mr,lr}^j \\ c_{lr,mr}^j & c_{lr,lr}^j \end{bmatrix} \]

\[ c_{l,g}^j = \frac{\sum_i^N Pr(R_i = g) * n_{il}^j}{\sum_i^N \sum_{k \in G} Pr(R_i = g) * n_{ik}^j} \tag{2} \]

Step 3: Document pair preference class. Assuming the labels observed from workers are totally independent, using the confusion matrix we obtained from the last step, for each document pair \((d1, d2)\) we estimate the preference relevance probability that each document pair being \(mr\) or \(lr\):

\[ Pr(R_i = mr | \forall j \in M) = \frac{pmr * \prod_{j=1}^M \prod_{l=0}^G (c_{lr,lr}^j)^{n_{il}^j}}{pmr * \prod_{j=1}^M \prod_{l=0}^G (c_{lr,lmr}^j)^{n_{il}^j} + prlr * \prod_{j=1}^M \prod_{l=0}^G (c_{lr,lr}^j)^{n_{il}^j}} \tag{3} \]

\[ Pr(R_i = n | \forall j \in M) = \frac{prlr * \prod_{j=1}^M \prod_{l=0}^G (c_{lr,lr}^j)^{n_{il}^j}}{pmr * \prod_{j=1}^M \prod_{l=0}^G (c_{lr,lmr}^j)^{n_{il}^j} + prlr * \prod_{j=1}^M \prod_{l=0}^G (c_{lr,lr}^j)^{n_{il}^j}} \tag{4} \]

14
Step 4: Repeat step 2 and step 3 until results converged. We repeat calculation of worker confusion matrices and estimation of document relevance grades until the result converge. We consider this process to have converged when, for each document $i$, the difference between $Pr(R_i = g), \forall g \in G$ at iteration $t - 1$ and at iteration $t$ is less than or equal to 0.01 for all $g$.

3.4.2 Experiments

We performed three groups of experiments, in which we varied the EM input data and the set of pair preference classes. In the first experiment group we only input preferences where the worker expressed either “more relevant” or “less relevant”, and the preference classes were accordingly restricted to $G = \{mr, lr\}$. Our second and third experiment groups consider the “Tie” judgments, but in different ways. In the second experiment group we input all preference judgements, but continue to restrict preference classes, $G = \{mr, lr\}$. Finally, in our third experiment group we input all three kinds of preference judgements and allowed the complete set of preference classes, $G = \{mr, lr, tie\}$.

The input data of first experiment group is less noisy than the other two because ambiguous judgements are omitted, however this may also lead to information loss. The third experiment group uses more noisy input data despite the more complex model.

Empirically, given a limited number of observations, the fewer free parameters in a mathematical model, the greater the reliability of its estimates. Accordingly, in different experiment groups we try to limit the number of free parameters to explore how this effects our results, however, we accept that a complex system may be the only approach which provides enough information.

In each experiment group we designed two experiments to explore how the number of free parameters effect the performance of our system. First, we perform a basic experiment with each group in which we calculate all the free parameters of the model. Next, we do a reduced-parameters experiment in which we decrease the number of free parameters according to the the requirements of experimental setup.

Experiment Group I
Input: In the first experiment group we only consider the preference judgments, omitting all the ties.
Preference Class Set: There are two preference classes, “more relevant” and “less relevant”.

Basic experiment: Each crowd worker has a $2 \times 2$ confusion matrix. In this experiment, we estimate all four parameters in the matrix.

$$G = \{mr, lr\}$$

$$M = \begin{bmatrix} \hat{c}_{mr, mr} & \hat{c}_{mr, lr} \\ \hat{c}_{lr, mr} & \hat{c}_{lr, lr} \end{bmatrix}$$

Reduced parameters: We sum a crowd worker’s true positive rate and true negative rate, sum his false positive rate and false negative rate, and normalize these to a distribution describing the probability
that the worker makes correct decisions or incorrect decisions. We repeat these values in the matrix.

\[ G = \{mr, lr\} \]

\[ c^j_{\text{correct}} = c_{mr, mr} = c_{lr, lr} \]

\[ c^j_{\text{incorrect}} = 1 - c^j_{\text{correct}} = c_{mr, lr} = c_{lr, mr} \]

\[ c^j_{\text{correct}} = \frac{\sum_i^N \Pr(R_i = mr) \times n_{mr}^j + \sum_i^N \Pr(R_i = lr) \times n_{lr}^j}{\sum_i^N \sum_k \Pr(R_i = mr) \times n_{mr, k}^j + \sum_i^N \sum_k \Pr(R_i = lr) \times n_{lr, k}^j} \]

\[ M = \begin{bmatrix} c_{\text{correct}} & c_{\text{incorrect}} \\ c_{\text{incorrect}} & c_{\text{correct}} \end{bmatrix} \]

**Experiment Group II:**

**Input:** In this experiment group we use all the judgments. For an arbitrary document pair \((doc_1, doc_2)\), \(doc_1\) may be more relevant than \(doc_2\), less relevant than \(doc_2\), or they may tie.

**Preference Class Set:** There are two preference classes, “more relevant” and “less relevant”.

**Basic experiment:** In this experiment, the worker quality confusion matrix is a 3 x 2 matrix. We estimate all 6 element in the matrix.

\[ G = \{mr, lr\} \]

\[ M = \begin{bmatrix} c_{mr, mr} & c_{mr, lr} \\ c_{lr, mr} & c_{lr, lr} \\ c_{tie, mr} & c_{tie, lr} \end{bmatrix} \]

**Reduced parameters:** Similarly to the transformation of the 2 x 2 matrix in Experiment Group I, we use \(c_{\text{correct}}\) and \(c_{\text{incorrect}}\) in the matrix. Using the same calculation, we assume the probability that the Turker labels a “more relevant” pair as “tie” to be the same as the probability that he labels a “less relevant” pair as “tie”.

\[ G = \{mr, lr\} \]

\[ c_{\text{correct}} = c_{mr, mr} = c_{lr, lr} \]

\[ c_{\text{pref}, ref} = c_{mr, lr} = c_{lr, mr} \]

\[ c_{tie, ref} = c_{tie, mr} = c_{tie, lr} \]
Experiment Group III:

Input: We use all the judgments in this experiment group. For an arbitrary document pair \((doc1, doc2)\), \(doc1\) may be more relevant than \(doc2\), less relevant than \(doc2\), or they may tie.

Preference Class Set: There are three preference classes; “more relevant”, “less relevant”, and “tie”.

Since the set of pair preference classes has changed, the initialization of the probabilities of different preference classes must change as well. The probability that a document pair is judged both “more relevant” and “less relevant” by workers is 0.35. The probability that a document pair is judged “tie” is 0.3.

For each document pair \((doc1, doc2)\), EM calculates the probabilities for “\(doc1\) is more relevant than \(doc2\)”, “\(doc1\) is less relevant than \(doc2\)”, and “\(doc1\) ties \(doc2\)”. We treat the probability of two documents being “Tie” half as “\(doc1\) is more relevant than \(doc2\)”, and the other half as “\(doc1\) is less relevant than \(doc2\)”. 

Basic experiment: In this experiment, the worker confusion matrix is 3 x 3. We estimate the all 9 elements in the matrix.

Reduced parameters: In addition to the assumptions made in the first and the second experiment groups, we assume the probability that a Turker labels a “more relevant” pair “tie” is the same as the probability that he labels a “less relevant” pair “tie”.

\[
G = \{mr, lr, tie\}
\]

\[
M = \begin{pmatrix}
c_{correct} & c_{pref,ref} \\
c_{pref,ref} & c_{correct} \\
c_{tie,ref} & c_{tie,ref}
\end{pmatrix}
\]

\[
G = \{mr, lr, tie\}
\]

\[
p_a = c_{mr, mr} = c_{lr, lr}
\]

\[
p_b = c_{mr, lr} = c_{lr, mr}
\]

\[
p_c = c_{tie, mr} = c_{tie, lr}
\]

\[
p_d = c_{mr, tie} = c_{lr, tie}
\]

\[
p_e = c_{tie, tie}
\]
Before we use Elo to transform the pair preference probabilities into a ranked document list, we use a histogram to compare the output from EM to the ground truth given by TREC in the qrel file. This file categorizes each document in each query set as either relevant or non-relevant to the query. To build the histogram, for each query we pick all the document pairs which include one relevant and one non-relevant document based on the qrel labels. For each pair, \((\text{doc}_1, \text{doc}_2)\), in which \(\text{doc}_1\) is not already more relevant than \(\text{doc}_2\), we swap their order and flip the estimated probability given by EM. Finally, we plot all the picked pairs on a histogram where the x-axis is the EM probability that \(\text{doc}_1\) is more relevant than \(\text{doc}_2\), and the y-axis is frequency.

From the pair judgement data for Experiment Group I, containing only judgements of “more relevant” and “less relevant”, we pick 11,236 pairs across all queries, which the qrel confirms are not “tie”. From the pair judgement data for Experiment Groups II and III, which contain all three judgement types, we pick 11,729 pairs across all queries, which the qrel confirms are not “tie”.

Data points (document pairs) for which EM predicts \(\text{doc}_1\) is more relevant than \(\text{doc}_2\) with greater than 0.5 probability now fall onto the right side of the histogram, showing agreement between EM and the qrel file. Points which EM predicts less than a 0.5 probability that \(\text{doc}_1\) is more relevant than \(\text{doc}_2\) fall on the left side of the histogram and show disagreement between EM and the qrel. The more document pairs on the right side of the histogram, the better the result.

Experiment Group I:

Figure 4 is the histogram of the first experiment group. The green line shows the result of basic experiment in which we estimated all the potential free parameters in the 2 x 2 worker quality confusion matrix. The red line shows the experiment in which we reduced the parameters. From this result we can see that there is more agreement between the EM result and the qrel file when we calculate less free parameters. Both experiments show that EM is able to correctly predict the majority of the picked pairs despite noisy input, however there are pairs for which EM opposed the qrel with extreme confidence.

The basic experiment matched the qrel file in 10,976 out of 11,236 (97.69%) pairs. In the reduced parameter experiment, 10,916 out of 11,236 (97.15%) pairs matched the preference label implied by the qrel file. These results are quite close. Figure 4 shows that the basic experiment produced fewer extreme opposite judgements, and more correct judgements with low-confidence, \(x = [0.5, 0.8]\). The reduced parameter experiment had more correct pairs with high confidence, but also had more pairs at the left tail of the histogram, which are extremely opposite of the qrel labels.
Experiment Group II:

Figure 5 shows the two experiments in the second group. The green line shows the result of the basic experiment in which we estimate all the 6 free parameters of the 3 x 2 worker quality confusion matrix. The red line shows the result of the reduced parameter version of experiment. This histogram has a similar shape as that of Group I. EM can predict most of the pairs correctly, but there are still some pairs for which it opposes the $qrel$ file.

The basic experiment matched the $qrel$ file for 11,355 out of 11,729 (96.81%) pairs. The reduced parameter experiment matched 11,387 out of 11,729 (97.08%) pairs. The results of these two experiments are also close. Similarly to the result of Group I, Figure 5 shows the basic experiment for Group II produced few preferences extremely opposed to the $qrel$, and more correct pairs pairs with low-confidence $x = [0.5, 0.75]$. The reduced parameter experiment produced more correct pairs with high-confidence, however, it also produced more incorrect pairs at the low-confidence tail.

The main differences between the first two groups are:

- The input data of Group I is less noisy. By removing the “Tie” data, we removed ambiguous and less informative data as well. Assuming Turkers were trying to give correct judgements, they would label
a pair as “Tie” only if the two documents seemed to have the same relevance or the two documents were confusing for the Turker.

- Group II has more input judgments and more noise than Group I. Since for Group I we did not include tie judgements, and some crowd workers only submitted a small number of judgements among which there were many ties, some workers may not have provided enough observations to estimate their quality accurately. Empirically, more than 10 times the number of parameters being estimated observations are required to accurate measure a worker’s quality, $n = 10 \times pc$.

**Experiment Group III:**

Figure 6 shows the results of the two experiments in Group III. The green line is the result of the basic experiment, in which we fully estimated all the free parameters in the model. The red line is the reduced parameter experiment, of which 4 parameters were estimated when we calculate the $3 \times 3$ confusion matrix. EM still produced correct preferences for the majority of the pairs. Since we added “tie” into the set of preference classes, every pair now has three probabilities, including, “more relevant” and “less relevant”. In this way, the result os EM are more precise for each pair.
The basic experiment matched 11,449 out of 11,729 (97.61%) pairs with the qrel file. The reduced parameter experiment matched 10,916 out of 11,420 (97.37%) pairs. The two experiments in Group III are close, but they are slightly better than the corresponding results in Group II. This may be a consequence of settling upon a more correct set of preference classes in the model than in Group II, despite working on the same noisy input.

Each of the experiments produced only approximate 1000 document pairs with extremely incorrect preferences. We tracked down some of the documents, here are the potential reasons lead Turkers make incorrect judgments.

**Restrictive Narratives** In our data pool, some of the queries have very restrictive narratives while others are broader. For example, for query 432 “Profiling motorists, police”, documents are relevant only if they discuss “police department criteria for identifying motorists considered likely to be carrying contraband”. Some Turkers may misread or misunderstand the narrative and treat all the documents including “profiling, motorists, police” keywords as relevant. For the queries with restrictive narratives, it is more difficult to correctly assess relevance.

On the other hand, some of the queries have more tolerant narratives. Query 447, “Stirling engine”, allows any document discussing development or applications of a *Stirling Engine*. For these queries, it is less
easy for the Turkers to make incorrect preference judgments.

**Not obvious information** Some of the documents are pages long, and are relevant because of one or two lines of relevant information. Our HITs require a lot of effort from each Turker. After pages of reading, it's quite easy for a Turker to miss out on a relevant bit of information and make an incorrect judgment.

Some queries are abstract, different people may have different opinions about the topic and disagree on document relevance. Query 417, “Creativity” includes definitions of creativity, descriptions of characteristics associated with creativity, and factors linked to creativity. This broad narrative about relevant documents may include some characteristics which some people view as creativity while others view them as old fashioned.

Figure 8: Example of relevant document which has bad rendering.

**Document Render Format** The data set includes some documents with very bad rendering. Figure 7 shows a relevant document for the “Stirling engine” query. It discusses how stirling cycle systems may be acceptable as an alternative technology to household freezers. However, since its bad rendering format makes it difficult to read, almost all the Turkers treat it as a non-relevant document when faced with it in a pair. If a strong majority of Turkers label this document as having less or equal relevance than all the other documents it makes pairs with, then EM would treat this document as non-relevant and punish any Turker that labeled it as relevant. In the histogram for this query, all the document pairs involving this document fell into the left side of the graph.
3.5 Phase II: Elo Rating System

Elo is a method for calculating the relative skill levels of players in two-player games such as chess. It assumes the performance of each player in each game is a normally distributed random variable, and the mean value of the performance of any given player changes only slowly over time. In this system, each player has a numerical rating inferred from wins, draws and losses. A higher rating indicates a better player, based on the game results against other players. Both players involved in a match gain or lose the same amount of points. The number of points gained or lost depends on the difference in the ratings of the two players. When winning against a player with a higher rating, more points are awarded. When winning against a player with a lower rating, few points are awarded. The system adjusts ratings for players after each game or group of games. If a player beats many other players with a higher rating, then his rating will go up. Otherwise, if a player loses to other players with a similar or lower rating, his rating will go down.

Elo also includes a straightforward way of estimating the expected number of wins and losses for each player. If player won more than he was expected to win, his rating would be adjusted upward. Likewise, if he lost more than was expected, his rating would be adjusted downward.

We use Elo to rank documents based on pair preferences. Elo treats all preferences as equally valid without considering worker quality, and so we use EM to filter out judgments given by low quality workers. EM estimates the quality of each worker, and ignores or flips low quality workers’ judgments. This may even lead to more accurate preferences than were input. Finally, we feed the output of EM into Elo to get a ranked list.

3.5.1 Algorithm details

As described above, document rankings can only be inferred by comparison and judgement against to other documents. However, we can estimate judgements based on the current value of their rankings, and then compare this estimation against the actual input judgements.

A player’s expected score is his probability of winning plus half his probability of drawing. For example, if the probability that player A beats player B is 0.7, the probability that A loses to B is 0.1, and probability a draw is 0.2, then the expected score of player A is $0.7 + 1/2 \times 0.2 = 0.8$. In Elo, a draw is considered as half a win and half a loss. If player A currently has rating $R_A$, and player B is rated $R_B$, the expected score of player A is:

$$E_A = \frac{1}{1 + 10^{(R_B - R_A)/400}} \quad (5)$$

The constant 400 shows that for each 400 rating points of advantage over player B, player A’s the chance of winning is magnified 10 times in comparison to player B’s chance of winning. If a player’s actual score after a match is higher then his expected score, Elo will adjust this player’s rating upward. If a player’s actual score after a match is lower than his expected score, Elo will adjust this player’s rating downward.
Assuming player A’s expected score is $E_A$ and after a match the actual score is $S_A$, we update player A’s rating with the following rule.

$$R_A' = R_A + K \times (S_A - E_A) \quad (6)$$

In practice the maximum possible adjustment per game was set at $K = 16$ for master chess players and $K = 32$ for other players. This adjustment can take place after one or more matches by summing expected and actual score across the matches. In our experiment we used $K = 20$ and updated a document’s ranking after every five judgements.

### 3.5.2 Elo Rating System Implementation

In order to produce a ranked list we feed the EM output to Elo. For each document pair in each query dataset, EM outputs a probability distribution over preference classes. For each query we initialize all the document ratings to 1000. We set the value of the constant $K$ to 20 for adjusting document ratings, and use 200 as the denominator for differences between the ratings of two documents when calculating expected document scores. We repeat Elo 15 times, each time using the previous run’s document ratings for initialization, and track how the accuracy of the ranked list changes.
Table 4 and Table 5 show the results of Elo after each iteration for the output of the basic experiment from Group I. A few iterations of Elo can improve the MAP of the ranked list, but little improvement is seen after 10 iterations. We run only 7 iterations of Elo for all other experiments.

4 Evaluation

4.1 Mean Average Precision and trec_eval

We evaluate ranked document lists primarily using Mean Average Precision (MAP). Trec_eval implements the calculations for MAP, and is also a standard evaluation tool used by the TREC community to provide a common measurement for different IR technologies. Trec_eval handles streams of documents, queries, and relevance judgments. Trec_eval outputs an evaluation for each query in the stream, as well as an overall evaluation. The output includes the number of documents under consideration, the number of relevant documents, the number of relevant documents which were retrieved in the given stream, precision at interpolated recall values [0, 0.1, 0.2, .. 1], mean average precision, precision at document-cutoffs [5, 10, 15, 20, 30, 100, 200, 500, 1000], and R-precision.

Precision is the proportion of a retrieved set that is relevant.

\[
Precision = \frac{|relevant \cap retrieved|}{|retrieved|} \tag{7}
\]

Recall is the proportion of all relevant documents in the collection included in the retrieved set.

\[
Recall = \frac{|relevant \cap retrieved|}{|relevant|} \tag{8}
\]

Average Precision is a way of considering both precision and recall, using one number to reflect the quality of entire list. It calculates the average of the precision values at eleven recall values [0, 0.1, ..0.9, 1].

The average is over all relevant documents and the relevant documents not retrieved get a precision score of zero.

Trec_eval evaluates IR systems by calculating their Mean Average Precision (MAP), which is the average of the precision value obtained for the top k documents, each time a relevant document is retrieved. MAP avoids interpolation or use of fixed recall levels. Trec_eval evaluates IR systems based on MAP.

\[
MAP = \frac{\sum_{q=1}^{Q} AvgP(q)}{Q} \tag{9}
\]

, Q is the query set.

Trec_eval takes a ranked results from IR systems sorted by their relevance score, the most relevant one has the highest score, and the least relevant one has the lowest score. Before we import our results we
truncating the ranked list into 1000 top retrieved documents. To run the script, we run 
_on the script, we run ./trec_eval [-q/-a] trec-qrel-file trec-result-file_. The qrel file has the following format: query-number 0 document-id relevance. The query-number is the query id. The document-id is the external ID for the judged documents. Relevance is the relevance assigned to the document for the particular query judged by the expert assessor. Relevance is a binary number, either 0 (non-relevant) or 1 (relevant).

The result-file has the following format: query-number Q0 document-id rank score Exp. The query-number is the identify number of the query. Q0 and Exp are constants that are used by some evaluation software. The document-id is the external ID for the retrieved document. The score is the score generated by the retrieval system for document document-id against the query query-number. -q is a parameter specifying detail for all queries. -a is a parameter specifying summary output only.

4.2 ROC Curve

Receiver Operating Characteristic curve (ROC) is a plot which aids in assessment of binary classifiers. It plots a True Positives Rate (TPR) against a False Positive Rate (FPR). TPR is also known as sensitivity. FPR is equal to 1 − specificity. ROC analysis provides a visual way to assess classifier correlation with examples, and helps in selection of a classifier score threshold.

\[
TPR = \text{sensitivity} = \frac{TP}{P} \\
FPR = 1 - \text{specificity} = \frac{FP}{N}
\]

TPR is the number of true positive results among all positive examples. FPR is the number of incorrect positive results among all negative examples. The x-axis of an ROC curve is the FPR and the y-axis is the TPR. The best possible model would yield a point in the upper left corner (0,1) of the ROC space, representing 100% sensitivity and 100% specificity.

4.3 Evaluation for Elo Ranking System

In this section we discuss the evaluations of different systems. We compare using just the Elo to the combination of EM and Elo.

4.3.1 Running Elo with / without “Tie” judgments

Table 6 and Table 7 show the results of running Elo over the crowdsourced judgments without any preprocessing. Table 6 explains the experiment run over all the judgments including the ties. Table 7
Table 6: Elo Rating System running on all judgments, including ties

<table>
<thead>
<tr>
<th></th>
<th>411</th>
<th>416</th>
<th>417</th>
<th>420</th>
<th>427</th>
<th>432</th>
<th>438</th>
<th>445</th>
<th>446</th>
<th>447</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAP</td>
<td>0.188</td>
<td>0.268</td>
<td>0.237</td>
<td>0.175</td>
<td>0.210</td>
<td>0.008</td>
<td>0.240</td>
<td>0.188</td>
<td>0.307</td>
<td>0.733</td>
<td>0.255</td>
</tr>
<tr>
<td>Retrieved</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>Relevant</td>
<td>27</td>
<td>42</td>
<td>75</td>
<td>33</td>
<td>50</td>
<td>28</td>
<td>173</td>
<td>62</td>
<td>162</td>
<td>16</td>
<td>668</td>
</tr>
<tr>
<td>Retrieved-relevant</td>
<td>21</td>
<td>42</td>
<td>70</td>
<td>32</td>
<td>47</td>
<td>13</td>
<td>153</td>
<td>56</td>
<td>148</td>
<td>16</td>
<td>598</td>
</tr>
<tr>
<td>Precision at 5 docs</td>
<td>0.40</td>
<td>0.60</td>
<td>0.60</td>
<td>0.40</td>
<td>0.80</td>
<td>0.0</td>
<td>0.80</td>
<td>0.80</td>
<td>1.0</td>
<td>0.80</td>
<td>0.62</td>
</tr>
<tr>
<td>Precision at 10 docs</td>
<td>0.5</td>
<td>0.30</td>
<td>0.30</td>
<td>0.20</td>
<td>0.50</td>
<td>0.0</td>
<td>0.70</td>
<td>0.40</td>
<td>0.80</td>
<td>0.80</td>
<td>0.45</td>
</tr>
<tr>
<td>Precision at 15 docs</td>
<td>0.33</td>
<td>0.267</td>
<td>0.20</td>
<td>0.267</td>
<td>0.40</td>
<td>0.0</td>
<td>0.533</td>
<td>0.267</td>
<td>0.733</td>
<td>0.80</td>
<td>0.38</td>
</tr>
<tr>
<td>Precision at 20 docs</td>
<td>0.3</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.35</td>
<td>0.0</td>
<td>0.40</td>
<td>0.35</td>
<td>0.65</td>
<td>0.65</td>
<td>0.33</td>
</tr>
<tr>
<td>Precision at 30 docs</td>
<td>0.233</td>
<td>0.233</td>
<td>0.367</td>
<td>0.20</td>
<td>0.233</td>
<td>0.0</td>
<td>0.40</td>
<td>0.33</td>
<td>0.60</td>
<td>0.433</td>
<td>0.303</td>
</tr>
</tbody>
</table>

Table 7: Elo Rating System running on judgments without ties

<table>
<thead>
<tr>
<th></th>
<th>411</th>
<th>416</th>
<th>417</th>
<th>420</th>
<th>427</th>
<th>432</th>
<th>438</th>
<th>445</th>
<th>446</th>
<th>447</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAP</td>
<td>0.238</td>
<td>0.345</td>
<td>0.295</td>
<td>0.192</td>
<td>0.335</td>
<td>0.017</td>
<td>0.282</td>
<td>0.245</td>
<td>0.318</td>
<td>0.816</td>
<td>0.3084</td>
</tr>
<tr>
<td>Retrieved</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>Relevant</td>
<td>27</td>
<td>42</td>
<td>75</td>
<td>33</td>
<td>50</td>
<td>28</td>
<td>173</td>
<td>62</td>
<td>162</td>
<td>16</td>
<td>668</td>
</tr>
<tr>
<td>Retrieved-relevant</td>
<td>23</td>
<td>42</td>
<td>73</td>
<td>33</td>
<td>49</td>
<td>17</td>
<td>154</td>
<td>57</td>
<td>145</td>
<td>16</td>
<td>609</td>
</tr>
<tr>
<td>Precision at 5 docs</td>
<td>0.40</td>
<td>0.60</td>
<td>0.60</td>
<td>0.40</td>
<td>1.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.80</td>
<td>1.0</td>
<td>0.80</td>
<td>0.66</td>
</tr>
<tr>
<td>Precision at 10 docs</td>
<td>0.50</td>
<td>0.30</td>
<td>0.60</td>
<td>0.40</td>
<td>0.70</td>
<td>0.0</td>
<td>0.80</td>
<td>0.40</td>
<td>0.90</td>
<td>0.90</td>
<td>0.55</td>
</tr>
<tr>
<td>Precision at 15 docs</td>
<td>0.40</td>
<td>0.267</td>
<td>0.533</td>
<td>0.267</td>
<td>0.467</td>
<td>0.0</td>
<td>0.60</td>
<td>0.433</td>
<td>0.80</td>
<td>0.80</td>
<td>0.447</td>
</tr>
<tr>
<td>Precision at 20 docs</td>
<td>0.35</td>
<td>0.30</td>
<td>0.550</td>
<td>0.20</td>
<td>0.40</td>
<td>0.0</td>
<td>0.60</td>
<td>0.35</td>
<td>0.75</td>
<td>0.75</td>
<td>0.425</td>
</tr>
<tr>
<td>Precision at 30 docs</td>
<td>0.333</td>
<td>0.333</td>
<td>0.50</td>
<td>0.233</td>
<td>0.33</td>
<td>0.067</td>
<td>0.50</td>
<td>0.40</td>
<td>0.633</td>
<td>0.533</td>
<td>0.387</td>
</tr>
</tbody>
</table>

explains the experiment run with only preference judgments. The input data of no-tie experiment is less noisy since the removal of all the unsure or equal relevance pairs. The no-tie experiment produced a higher MAP for all of the 10 queries.

### 4.4 Evaluation for EM and Elo Ranking System

We run Elo 7 times using EM preference class probabilities as input. This section shows the results of EM plus Elo for the three experiment groups.

#### 4.4.1 Experiment Group I

Figure 8 shows the ROC curve for all of the basic experiments across the 10 queries in Group I. Figure 9 shows the ROC curve of the reduced parameter experiments across the 10 queries. These two figures show that different queries got dramatically different results when run through the same model. Query 447 produced a an ideal ROC Curve, while query 432 looks almost random. Possible explanations include differences among the queries themselves, and noise in the crowdsourced data.
Figure 9: ROC Curve for basic experiment in Group I

Figure 10: ROC Curve for parameter cut experiment in Group I

<table>
<thead>
<tr>
<th>2x2</th>
<th>MAP</th>
<th>411</th>
<th>416</th>
<th>417</th>
<th>420</th>
<th>427</th>
<th>432</th>
<th>438</th>
<th>445</th>
<th>446</th>
<th>447</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Retrieved</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>10000</td>
</tr>
<tr>
<td></td>
<td>Relevant</td>
<td>27</td>
<td>42</td>
<td>75</td>
<td>33</td>
<td>50</td>
<td>28</td>
<td>173</td>
<td>62</td>
<td>162</td>
<td>16</td>
<td>668</td>
</tr>
<tr>
<td></td>
<td>Retrieved-relevant</td>
<td>25</td>
<td>41</td>
<td>68</td>
<td>32</td>
<td>48</td>
<td>18</td>
<td>138</td>
<td>59</td>
<td>135</td>
<td>16</td>
<td>580</td>
</tr>
<tr>
<td></td>
<td>Precision at 5 docs</td>
<td>0.40</td>
<td>0.60</td>
<td>1.0</td>
<td>0.20</td>
<td>0.800</td>
<td>0.0</td>
<td>0.800</td>
<td>0.40</td>
<td>1.0</td>
<td>1.0</td>
<td>0.620</td>
</tr>
<tr>
<td></td>
<td>Precision at 10 docs</td>
<td>0.60</td>
<td>0.50</td>
<td>0.80</td>
<td>0.20</td>
<td>0.600</td>
<td>0.0</td>
<td>0.600</td>
<td>0.60</td>
<td>0.80</td>
<td>1.0</td>
<td>0.570</td>
</tr>
<tr>
<td></td>
<td>Precision at 15 docs</td>
<td>0.667</td>
<td>0.53</td>
<td>0.80</td>
<td>0.20</td>
<td>0.533</td>
<td>0.067</td>
<td>0.533</td>
<td>0.60</td>
<td>0.867</td>
<td>0.933</td>
<td>0.573</td>
</tr>
<tr>
<td></td>
<td>Precision at 20 docs</td>
<td>0.550</td>
<td>0.50</td>
<td>0.750</td>
<td>0.20</td>
<td>0.550</td>
<td>0.050</td>
<td>0.500</td>
<td>0.60</td>
<td>0.80</td>
<td>0.750</td>
<td>0.525</td>
</tr>
<tr>
<td></td>
<td>Precision at 30 docs</td>
<td>0.40</td>
<td>0.467</td>
<td>0.60</td>
<td>0.20</td>
<td>0.567</td>
<td>0.033</td>
<td>0.433</td>
<td>0.567</td>
<td>0.70</td>
<td>0.533</td>
<td>0.450</td>
</tr>
</tbody>
</table>

Table 8: Results for Elo Ranking System run on basic from experiment group I

Table 8 and Table 9 are the trec_eval evaluation for the experiments combining EM and Elo run for the Group I experiments. Table 8 is for the basic experiment, and table Table 9 is for the reduced parameter experiment. The basic experiment produced a higher MAP for the 10 queries in general.

In the reduced parameter experiment, when we used just the correctness probability and the incorrectness probability, and since \( Pr(\text{correct}) = 1 - Pr(\text{incorrect}) \), there is really only 1 parameter in the confusion matrix to estimate. Based on our HITs design, we guarantee at least 15 judgments from each Turker, which is more than 10 times of the number of parameters. Therefore, relative to the observations collected, the reduced parameter model is oversimplified.

Table 10 shows a comparison of running pure Elo and running EM plus Elo. The experiment combining basic EM and Elo got higher MAP value than using Elo only as we expected.

4.4.2 Experiment Group II
<table>
<thead>
<tr>
<th>2x2</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAP</td>
<td>Retrieved</td>
<td>Relevant</td>
<td>Retrieved-relevant</td>
<td>Precision at 5 docs</td>
<td>Precision at 10 docs</td>
<td>Precision at 15 docs</td>
<td>Precision at 20 docs</td>
<td>Precision at 30 docs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>411</td>
<td>0.191</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>416</td>
<td>0.331</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>417</td>
<td>0.241</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>420</td>
<td>0.145</td>
<td>50</td>
<td>28</td>
<td>173</td>
<td>62</td>
<td>162</td>
<td>16</td>
<td>668</td>
<td>479</td>
</tr>
<tr>
<td>427</td>
<td>0.398</td>
<td>0.20</td>
<td>0.10</td>
<td>0.20</td>
<td>0.13</td>
<td>0.40</td>
<td>43</td>
<td>0.36</td>
<td>0.264</td>
</tr>
<tr>
<td>432</td>
<td>0.013</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.1</td>
<td>0.0</td>
<td>0.52</td>
<td>0.385</td>
<td>0.327</td>
</tr>
<tr>
<td>438</td>
<td>0.152</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.385</td>
<td>0.327</td>
</tr>
<tr>
<td>445</td>
<td>0.100</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.385</td>
<td>0.327</td>
</tr>
<tr>
<td>446</td>
<td>0.051</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.385</td>
<td>0.327</td>
</tr>
<tr>
<td>447</td>
<td>0.099</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.385</td>
<td>0.327</td>
</tr>
<tr>
<td>Average</td>
<td>0.2613</td>
<td>0.52</td>
<td>0.43</td>
<td>0.40</td>
<td>0.36</td>
<td>0.38</td>
<td>0.38</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 11: Results for Elo Ranking System running on basic experiment from group II

<table>
<thead>
<tr>
<th></th>
<th>Elo</th>
<th>Basic EM plus Elo</th>
</tr>
</thead>
<tbody>
<tr>
<td>3x2</td>
<td>0.3084</td>
<td>0.358</td>
</tr>
</tbody>
</table>

Table 10: MAP values for EM plus Elo system and the Elo system

Figure 10 and Figure 11 show that different queries have different results running through the same model, and most queries got better performance in the basic experiments than the reduced parameter experiments.

Table 11 and Table 12 are the evaluations for the experiments combining EM and Elo run over experiment Group II. The basic experiment got better results than the parameter reduced experiments for majority queries. The basic experiment has 6 parameters in worker’s confusion matrix, however since the sum of a worker gives judgment to a pair with a certain relevance grade is 1. There are 4 parameters to be estimated. In the parameter reduced experiment, we reduced the number of parameters into 2. Since the same reason as above, \( c_{mr,mr} + c_{tie,lr} + c_{ls,lr} = 1 \) and \( c_{mr,lr} + c_{tie,lr} + c_{ls,lr} = 1 \). The result shows the parameter reduced model also oversimplified the system.

### 4.4.3 Experiment Group III

Figure and Figure are the ROC Curves for the experiments in Group III. Different queries have different
Figure 11: ROC Curve for basic experiment in Group II

Figure 12: ROC Curve for parameter cut experiment in Group II

<table>
<thead>
<tr>
<th></th>
<th>411</th>
<th>416</th>
<th>417</th>
<th>420</th>
<th>427</th>
<th>432</th>
<th>438</th>
<th>445</th>
<th>446</th>
<th>447</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAP</td>
<td>0.189</td>
<td>0.312</td>
<td>0.237</td>
<td>0.144</td>
<td>0.393</td>
<td>0.014</td>
<td>0.150</td>
<td>0.096</td>
<td>0.050</td>
<td>0.983</td>
<td><strong>0.257</strong></td>
</tr>
<tr>
<td>Retrieved</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>10000</td>
<td></td>
</tr>
<tr>
<td>Relevant</td>
<td>27</td>
<td>42</td>
<td>75</td>
<td>33</td>
<td>50</td>
<td>28</td>
<td>173</td>
<td>62</td>
<td>162</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>Retrieved-relevant</td>
<td>27</td>
<td>41</td>
<td>54</td>
<td>32</td>
<td>48</td>
<td>17</td>
<td>110</td>
<td>59</td>
<td>76</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>Precision at 5 docs</td>
<td>0.60</td>
<td>0.60</td>
<td>0.80</td>
<td>0.40</td>
<td>0.80</td>
<td>0.0</td>
<td>0.80</td>
<td>0.20</td>
<td>0.0</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>Precision at 10 docs</td>
<td>0.40</td>
<td>0.40</td>
<td>0.70</td>
<td>0.30</td>
<td>0.60</td>
<td>0.0</td>
<td>0.50</td>
<td>0.20</td>
<td>0.0</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>Precision at 15 docs</td>
<td>0.333</td>
<td>0.40</td>
<td>0.667</td>
<td>0.20</td>
<td>0.40</td>
<td>0.0</td>
<td>0.533</td>
<td>0.20</td>
<td>0.13</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>Precision at 20 docs</td>
<td>0.350</td>
<td>0.40</td>
<td>0.650</td>
<td>0.30</td>
<td>0.450</td>
<td>0.05</td>
<td>0.400</td>
<td>0.20</td>
<td>0.10</td>
<td>0.75</td>
<td></td>
</tr>
<tr>
<td>Precision at 30 docs</td>
<td>0.267</td>
<td>0.450</td>
<td>0.467</td>
<td>0.20</td>
<td>0.60</td>
<td>0.033</td>
<td>0.267</td>
<td>0.133</td>
<td>0.067</td>
<td>0.533</td>
<td>0.290</td>
</tr>
</tbody>
</table>

Table 12: Results for Elo Ranking System running on Parameter Reduced from experiment group II

<table>
<thead>
<tr>
<th></th>
<th>411</th>
<th>416</th>
<th>417</th>
<th>420</th>
<th>427</th>
<th>432</th>
<th>438</th>
<th>445</th>
<th>446</th>
<th>447</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAP</td>
<td>0.339</td>
<td>0.431</td>
<td>0.356</td>
<td>0.134</td>
<td>0.469</td>
<td>0.024</td>
<td>0.233</td>
<td>0.076</td>
<td>0.313</td>
<td>0.644</td>
<td><strong>0.329</strong></td>
</tr>
<tr>
<td>Retrieved</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>10000</td>
<td></td>
</tr>
<tr>
<td>Relevant</td>
<td>27</td>
<td>42</td>
<td>75</td>
<td>33</td>
<td>50</td>
<td>28</td>
<td>173</td>
<td>62</td>
<td>162</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>Retrieved-relevant</td>
<td>27</td>
<td>42</td>
<td>70</td>
<td>30</td>
<td>48</td>
<td>19</td>
<td>121</td>
<td>18</td>
<td>127</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>Precision at 5 docs</td>
<td>0.40</td>
<td>0.60</td>
<td>0.80</td>
<td>0.40</td>
<td>1.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.00</td>
<td>0.80</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>Precision at 10 docs</td>
<td>0.60</td>
<td>0.70</td>
<td>0.80</td>
<td>0.30</td>
<td>0.80</td>
<td>0.0</td>
<td>0.80</td>
<td>0.10</td>
<td>0.80</td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td>Precision at 15 docs</td>
<td>0.733</td>
<td>0.60</td>
<td>0.667</td>
<td>0.20</td>
<td>0.533</td>
<td>0.067</td>
<td>0.667</td>
<td>0.067</td>
<td>0.80</td>
<td>0.6</td>
<td></td>
</tr>
<tr>
<td>Precision at 20 docs</td>
<td>0.650</td>
<td>0.55</td>
<td>0.70</td>
<td>0.25</td>
<td>0.550</td>
<td>0.05</td>
<td>0.60</td>
<td>0.05</td>
<td>0.80</td>
<td>0.55</td>
<td></td>
</tr>
<tr>
<td>Precision at 30 docs</td>
<td>0.467</td>
<td>0.50</td>
<td>0.567</td>
<td>0.20</td>
<td>0.70</td>
<td>0.067</td>
<td>0.467</td>
<td>0.033</td>
<td>0.667</td>
<td>0.433</td>
<td></td>
</tr>
</tbody>
</table>

Table 13: Results for Elo Ranking System running on basic from experiment group III

Table 13 and Table 14 are the evaluation for the experiments combining EM and Elo run over experiment Group III. The reduced parameter experiment got better results than the basic experiments in general. In the basic experiments there are 9 raw parameters to be estimated. However, for each worker the sum of the probability that this worker gives judgments to a document pair of which doc1 is more relevant than
Figure 13: ROC Curve for basic experiment in Group III

Figure 14: ROC Curve for parameter cut experiment in Group III

<table>
<thead>
<tr>
<th>3x3</th>
<th>411</th>
<th>416</th>
<th>417</th>
<th>420</th>
<th>427</th>
<th>432</th>
<th>438</th>
<th>445</th>
<th>446</th>
<th>447</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAP</td>
<td>0.219</td>
<td>0.397</td>
<td>0.250</td>
<td>0.145</td>
<td>0.489</td>
<td>0.015</td>
<td>0.149</td>
<td>0.388</td>
<td>0.058</td>
<td>0.996</td>
<td><strong>0.3105</strong></td>
</tr>
<tr>
<td>Retrieved</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>10000</td>
</tr>
<tr>
<td>Relevant</td>
<td>27</td>
<td>42</td>
<td>75</td>
<td>33</td>
<td>50</td>
<td>28</td>
<td>173</td>
<td>62</td>
<td>162</td>
<td>16</td>
<td>668</td>
</tr>
<tr>
<td>Retrieved-relevant</td>
<td>27</td>
<td>41</td>
<td>53</td>
<td>32</td>
<td>47</td>
<td>18</td>
<td>109</td>
<td>59</td>
<td>78</td>
<td>16</td>
<td>480</td>
</tr>
<tr>
<td>Precision at 5 docs</td>
<td>0.60</td>
<td>0.60</td>
<td>0.80</td>
<td>0.20</td>
<td>1.0</td>
<td>0.0</td>
<td>0.60</td>
<td>0.60</td>
<td>0.0</td>
<td>1.0</td>
<td>0.540</td>
</tr>
<tr>
<td>Precision at 10 docs</td>
<td>0.40</td>
<td>0.50</td>
<td>0.80</td>
<td>0.30</td>
<td>0.80</td>
<td>0.0</td>
<td>0.50</td>
<td>0.70</td>
<td>0.10</td>
<td>1.0</td>
<td>0.510</td>
</tr>
<tr>
<td>Precision at 15 docs</td>
<td>0.40</td>
<td>0.533</td>
<td>0.733</td>
<td>0.20</td>
<td>0.667</td>
<td>0.0</td>
<td>0.40</td>
<td>0.733</td>
<td>0.133</td>
<td>1.0</td>
<td>0.480</td>
</tr>
<tr>
<td>Precision at 20 docs</td>
<td>0.35</td>
<td>0.55</td>
<td>0.70</td>
<td>0.25</td>
<td>0.65</td>
<td>0.5</td>
<td>0.40</td>
<td>0.75</td>
<td>0.10</td>
<td>0.8</td>
<td>0.460</td>
</tr>
<tr>
<td>Precision at 30 docs</td>
<td>0.233</td>
<td>0.40</td>
<td>0.50</td>
<td>0.233</td>
<td>0.667</td>
<td>0.033</td>
<td>0.30</td>
<td>0.633</td>
<td>0.10</td>
<td>0.533</td>
<td>0.363</td>
</tr>
</tbody>
</table>

Table 14: Results for Elo Ranking System running on reduced parameter from experiment group III

doc2, is 1. $c_{mr, mr} + c_{tie, mr} + c_{ls, mr} = 1$ and $c_{mr, lr} + c_{tie, lr} + c_{ls, lr} = 1$. There are 6 left to be estimated.
In the reduced parameter experiment we shrink the number of parameters to 3. Since some Turkers only submit one assignment, we can only guarantee that we got at least 15 judgments from each Turker, which is 2.5 times of the number of parameters in the basic experiment and 5 times of the number of parameters in the reduced parameter experiment. The result shows 3 parameters can keep a proper system complexity. We don’t have enough judgments to estimate 6 parameters. From comparing the results from Group II and Group III, 3 preference classes can get a more result than two preference classes.

### 4.5 Evaluation for Individual Queries

#### 4.5.1 Evaluation for Query 411

Figure 14 shows the results of query 411 using different models. All the results of the basic experiments are better than the reduced parameter experiments. The reduced parameter experiments perform similar to each other among all three groups. For the basic experiments the 3x3 confusion matrix works better than 3x2 confusion matrix, and better than the 2x2 confusion matrix. 191 Turkers labeled this query, which
means on average there are around 215 judgments from each Turker. This is enough to estimate all the parameters in any of our confusion matrices. Hence there is no necessity to reduce the complexity of the system, and the reduced parameter experiments may even over simplify the model.

### 4.5.2 Evaluation for Query 416

![ROC Curve for Query 416](image)

Figure 16: ROC Curve for Query 416

Figure 15 shows the results of query 416 using different models. This Figure shows both experiments in Group III perform better than the two experiments in Group I, and the experiments in Group I perform better than the experiments in Group II. The parameter reduced experiment in Group III reaches a TPR of 1 after the FPR hit 0.2, meaning the system quickly retrieved all the 42 relevant documents before we treat more than 20% of the non-relevant documents as relevant. All the systems do well for this query. The average number of judgments for each crowd worker in this query is 117.6. The query topic is “Three
gorges project”, the narrative shows “A relevant document will provide the projected date of completion of the project, its estimated total cost, or the estimated electrical output of the the finished project.” The narrative is precise and clear, which also helps crowd workers to make the correct judgments.

4.5.3 Evaluation for Query 417

Figure 17: ROC Curve for Query 417

Figure 16 shows the results of query 417 using different models. The basic experiment of Group I and Group III outperform the other experiments. But all the experiments have similar ROC curves. In Group I, the basic experiment outperformed the reduced parameter experiment before FPR achieved 0.6, but then it stopped retrieving relevant document until its FPR hit 0.8. The basic experiment of Group II did better when the FPR is less than 0.45. The basic experiment of Group III did better when the FPR is less than 0.5, and the performance of two experiments in Group III are similar. All the experiments retrieved relevant documents at the end of their ranked lists. The query topic is “creativity” with narrative “Relevant items include definitions of creativity, descriptions of characteristics associated with creativity, and factors linked to creativity.” This query is very subjective. Different crowd workers may have different ideas about creativity, and it is difficult to overcome the noise and the judgments.

4.5.4 Evaluation for Query 420

Figure 17 shows the results of query 420 using different models. The performances of all the models are similar. Only the basic experiment in Group II outperforms the reduced parameter experiment. Parameter reduced experiments did better in the other two groups. This query is about “Carbon monoxide poisoning”, with narrative “Relevant documents will contain data on what carbon monoxide poisoning is, symptoms, causes, and/or prevention. Advertisements for carbon monoxide protection products or services are not relevant. Discussions of auto emissions and air pollution are not relevant even though they can contain carbon monoxide.”
4.5.5 Evaluation for Query 427

Figure 18 shows the results of query 427 using different models. For this query, majority models got well and similar ROC curves except the basic experiment of Group Two and the basic experiment of Group Three. Those two basic experiments have most parameters to estimate.

4.5.6 Evaluation for Query 432

Figure 19 shows the results of query 432 using different models. All the models produced ROC curves which indicate that the ranked list is almost random. The narrative for Query 432 (Profiling, motorists, police) is very strict. “Description: Do police departments use “profiling” to stop motorists? Narrative: A relevant document will report or discuss police department criteria for identifying motorists considered likely
to be carrying contraband. Documents discussing the detention of individuals by foreign security forces are not relevant. ” Crowd workers are prone to give incorrect answer, because most of them are not native Americans, they may not familiar with the criteria of American police department. They may also treat documents discussing foreign countries’ criteria as relevant.

4.5.7 Evaluation for Query 438

Figure 20 shows the results of query 438 using different models. All the models produced competitive results, but the basic experiments of the three groups all outperformed their corresponding reduced parameter experiments for FPR less than 0.5. As the FPR increases, the models become indistinguishable. The query title is “Tourism, increase”. Relevant documents have to include a country name which has experienced an increase in tourism, the increase must represent the nation as a whole and tourism in general. We collected
43981 judgments for 1798 documents in total. The query is closer to crowd workers’ life experience and easy
for them to understand, unlike the technical query about the “stirling engine” or the subjective query about
“creativity”. This is the reason why this query got generally good MAP in all the models.

4.5.8 Evaluation for Query 445

Figure 22: ROC Curve for Query 445

Figure 21 shows the results of query 445 using different models. The curves split into two branches for this
query. One branch has TPR greater than 0.8 before FPR hits 0.3, the other branch performs slightly better
than randomly ranking. The query is about “women clergy”. Relevant documents must indicate either a
country where a woman has been installed as clergy or a country that is considering such an installation.
The basic experiment of Group I and the parameter reduced parameter experiment of Group III outperform
all the other models.

4.5.9 Evaluation for Query 446

Figure 22 shows the results of query 446 using different models. The curves of this query also branched
into two groups. The basic experiments of all three groups are in the better performing branch, while the
reduced parameter experiments are in the poorly performing branch. The basic experiments have average
MAP 0.3173, however the reduced parameter experiments have average MAP 0.053. The reduced parameter
experiment dramatically simplified the system complexity in each group. The query is about tourists and
violence. Relevant documents must discuss “accounts of known harm to tourists, single evidence or isolated
incidents are not relevant”. The query itself is clear. But since it is strict, crowd workers may easily treat
nonrelevant documents as relevant and make incorrect judgments.
4.5.10 Evaluation for Query 447

Figure 23 shows the results of query 447 using different models. The majority of the curves in this query are almost perfect, and reach the top left corner. The basic experiments of Group II and Group III got worse results than all the other models because these two models estimate the greatest number of parameters. We have on average 222 judgments per crowd worker. This is enough to estimate a small number of parameters, however the result drops when relying on too many estimates.

5 Conclusion

After analysis of each query and a comparison of all the models, it seems that running less-noisy data through the model which uses both EM and Elo produces the best results among all the other experiments.
we performed. The experiments which exclude “tie” judgements and use EM to estimate all the parameters in a 2x2 confusion matrix produce the best average-MAP over the ten queries. This experimental setup also produced the best MAP among models for the ten individual queries. Among the experiments which do not exclude “tie” judgements, the basic experiment of Group III which estimates 6 parameters produced the best average-MAP over the ten queries and the most best MAP among models for the ten individual queries.

Even though EM considers worker quality, if the collected data is too noisy then EM cannot assess which preferences are more correct. When using reasonably noisy data, EM can be utilized to filter out low quality crowd workers and will EM generate correct pair preferences for most of the document pairs. It is also important to keep a reasonable level of model complexity in order to retrieve enough information from the available data.
References


