The University of Padua (IMS) at TREC 2016 Total Recall Track

Giorgio Maria Di Nunzio, Maria Maistro, and Daniel Zilio

Dept. of Information Engineering – University of Padua
[dinunzio,maistro]@dei.unipd.it,daniel.zilio@unipd.it

Abstract. The participation of the Information Management System (IMS) Group of the University of Padua in the Total Recall track at TREC 2016 consisted in a set of fully automated experiments based on the two-dimensional probabilistic model. We trained the model in two ways that tried to mimic a real user, and we compared it to two versions of the BM25 model with different parameter settings. This initial set of experiments lays the ground for a wider study that will explore a gamification approach in the context of high recall situations.

1 Introduction

The principal purpose of the Total Recall Track is to evaluate, through a controlled simulation, methods to achieve very high recall (close to 100%) with a human assessor in the loop. The task to be solved is the following: given a topic description, identify the documents in a corpus, one at a time or in batches, such that all relevant documents are identified before all non-relevant documents, as nearly as possible.

The Information Management Systems (IMS) research group of the University of Padua participated for the first time in one of the two sub-tasks available, namely the "At Home" with the "athome4" collection. This collection consists of 290,000 Jeb Bush email messages and a set of 34 topics. A subset of emails sampled using the continuous active learning method [2] was labeled by a primary assessor as "relevant" or "not relevant". Relevant documents were further labeled by the primary assessor as "important" or "not important" and categorized into sub-categories corresponding to different subtopics or aspects of relevance. During the experiments, the information made available to the participants for each submitted document was either "relevant" or "not relevant".

In this paper, we present the experiments we carried out using a fully automated system based on the two-dimensional interpretation of the BM25 model [5]. We implemented two versions of this system that mimic the behaviour of two users that try to find the optimal decision with two different strategies. We also run two runs using a "plain" BM25 with different parameters. The results of this first set of experiments will lay the ground for our current work on the study of gamification of classification problems that will directly involve users in the problem of high recall systems.

2 Models

In this section, we present the models that we used to produce the four runs: the BM25 used as a baseline and the two-dimensional representation of the BM25.

2.1 BM25

For the BM25, we used the definition given by Zaragoza and Robertson in [7] where the weight of the i-th term in a document is equal to:

$$w_i^{BM25}(tf) = \frac{tf}{k_1\left((1-b)+b\frac{dl}{avdl}\right)+tf} w_i^{BIM} \tag{1}$$

where k_1 and b are two parameters (we used the default values used by Terrier[1], $k_1 = 1.2$ and $b = 0.75$), tf is the term frequency in the document, and w_i^{BIM} is the Binary Independence Model weight of the i-th term:

$$w_i^{BIM} = \log \frac{\theta_i^{\mathcal{R}}}{(1-\theta_i^{\mathcal{R}})} \frac{(1-\theta_i^{\mathcal{NR}})}{\theta_i^{\mathcal{NR}}} \tag{2}$$

where $\theta_i^{\mathcal{R}}$ and $\theta_i^{\mathcal{NR}}$ are the parameters of the Bernoulli random variable that represent the presence (or absence) of the i-th term in the relevant ($\mathcal{R}$) and non-relevant ($\mathcal{NR}$) documents. The estimate of each parameter is:

$$\theta_i^{\mathcal{R}} = \frac{r_i + \alpha^{\mathcal{R}}}{R + \alpha^{\mathcal{R}} + \beta^{\mathcal{R}}} \tag{3}$$

$$\theta_i^{\mathcal{NR}} = \frac{n_i - r_i + \alpha^{\mathcal{NR}}}{N - R + \alpha^{\mathcal{NR}} + \beta^{\mathcal{NR}}} \tag{4}$$

where R is the number of relevant documents, r_i the number of relevant documents in which the i-th term appears, N is the total number of documents and n_i is the total number of documents in which the i-th term appears. Parameters α and β correspond to the hyper-parameter of the conjugate beta prior distribution of the Bernoulli random variable. For $\alpha^{\mathcal{R}} = \beta^{\mathcal{R}} = 0.5$ and $\beta^{\mathcal{R}} =^{\mathcal{NR}}= 0.5$, we obtain the definition of the well-known Robertson - Spärck Jones weight w_i^{RSJ} [7].

2.2 Two-Dimensional Model

The two-dimensional representation of probabilities [3, 8], is an intuitive way of presenting a two-class classification problem on a two-dimensional space. Given two classes, for example $\mathcal{R}$ and $\mathcal{NR}$, a document d is assigned to category $\mathcal{R}$ if the following inequality holds:

$$\underbrace{P(d|\mathcal{NR})}_{y} < m\,\underbrace{P(d|\mathcal{R})}_{x} + q \tag{5}$$

[1] `http://terrier.org`

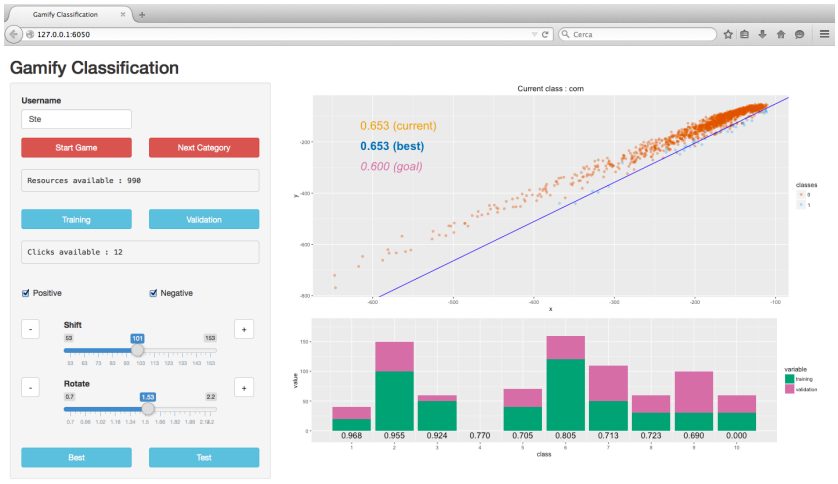


Fig. 1: Layout of the original "classification game" ([6]) that was adapted to the Total Recall track experiments.

where $P(d|\mathcal{R})$ and $P(d|\mathcal{NR})$ are the likelihoods of the object d given the two categories, while m and q are two parameters that can be assigned (automatically or by a user) to compensate for either the unbalanced classes situations or different misclassification costs.

If we interpret the two likelihoods as two coordinates x and y of a two dimensional space, the problem of classification can be studied on a two-dimensional plot. The decision of the classification is represented by the line $y = mx + q$ that splits the plane into two parts: all the points that fall 'below' this line are classified as objects that belong to class $\mathcal{R}$ (see Figure 1 for an example). Without entering into the mathematical details of this approach [3], the basic idea is that the two parameters m and q can be optimized by a user or by an automatic approach to obtain a better separation of the points.

Two-dimensional BM25 In order to link the two-dimensional model to the BM25 model, first we define the BIM weight as a difference of logarithms:

$$w_i^{BIM} = \log \frac{\theta_i^{\mathcal{R}}}{(1-\theta_i^{\mathcal{R}})} - \log \frac{\theta_i^{\mathcal{NR}}}{(1-\theta_i^{\mathcal{NR}})} = w_i^{BIM,\mathcal{R}} - w_i^{BIM,\mathcal{NR}} \tag{6}$$

then, we can define the BM25 term weight accordingly

$$w_i^{BM25}(tf) = \frac{tf}{k_1\left((1-b)+b\frac{dl}{avdl}\right)+tf}\left(\ w_i^{BIM,R} - \ w_i^{BIM,NR}\right) \tag{7}$$

We now have all the elements to define the two coordinates $x = P(d|\mathcal{R})$ and $y = P(d|\mathcal{NR})$ in the following way:

$$P(d|\mathcal{R}) = \sum_{i \in d} w_i^{BM25,\mathcal{R}}(tf) \tag{8}$$

$$P(d|\mathcal{NR}) = \sum_{i \in d} w_i^{BM25,\mathcal{NR}}(tf) \tag{9}$$

where $\sum_{i \in d}$ indicates (with an abuse of notation) the sum over all the terms of document d.

3 Experiments

In the previous section, we presented the mathematical framework of the two-dimensional framework and we showed that there are actually four parameters that can be used to optimize the classification (or retrieval) performance: α, β, m, and q. In our experiments, we fixed α and β and tweak m and q. There is also another issue related to the problem of finding a set of documents to be used as a training set in order to estimate $P(d|\mathcal{R})$ and $P(d|\mathcal{NR})$. We tackled this problem by iteratively identifying a potential training set of documents to be submitted to the "assessor" and use the new relevance information to find a better decision line. We repeat this process until we find a line (Equation 5) that allows us to obtain a recall on the already judged document greater than some 0.999 (this threshold can be optimized as well).

We submitted four runs: two baseline runs of BM25 with different smoothing parameters, one aggressive (**baseline_bm25_highly**) and one mild (**baseline_bm25_smoothing**); two two-dimensional runs with different strategies to send batches of documents (**auto_shift_rotate_exp** and **auto_shift_rotation**). In Appendix A, we show the pseudo-code used to find the potentially relevant documents for each run.

Data pre-processing Each email was pre-processed in order to have only the textual content (i.e., not the FROM, TO fields, nor the URLs or links at the bottom of the page); then, standard text processing was performed for all the four runs: all letters converted to lowercase, punctuations and numbers were removed, a stoplist of 174 terms was applied (we used the one included in the 'tm' R package [4]), words were stemmed by means of a Snowball stemmer ('SnowballC' R package [1]). A maximum of 50,000 features were selected at each round according to the difference $\theta_i^R - \theta_i^{NR}$.

4 Results

In this section, we briefly analyze some of results on four topics that we found representatives in terms of the range of behaviour of the best two runs, namely

'automatic_shift_rotate' and 'baseline_bm25_smoothing', compared to the Baseline Model Implementation (bmi) run provided by the organizers of the track. We could group the performance of the system into four distinct categories:

- The system performed well and tracked the performance of the bmi. See for example Figure 3a for the topic "athome402".
- The system performed well and the decision to "call the shot" early worked very well like, for example, the topic "athome424" as shown in Figure 3b.
- When the number of initial relevant documents is small (we fixed a low threshold for the maximum number of relevant documents within the 100 top ranked, see Appendix A), the system tend to miss a lot of relevant information due to the strong initial bias and it takes some effort, in terms of documents to assess, to recover this wrong start like, for example, topic 'athome403' in Figure 4a.
- The worst case is a negative start that continues with a negative feedback (only non relevant documents) that cannot recover the initial situation. See for example topic 'athome426' shown in Figure 4b.

At the end, we were positively surprised to see that in some cases this approach performed well and sometimes close to the bmi. These results are encouraging if we consider that i) our system is designed to be used by users who can provide feedback during the choice of the decision line, and ii) that it was the first time that we tried to implement an automatic strategy to find the best decision line. In confirmation of this intuition, we ran a quick manual search of the best decision line on topic 'athome403' (see Figure 2 for the actual interface) and we were able to find about 92% of relevant documents within the first 1,363 documents while the automatic approach arrived to the same ratio of relevant documents after judging 8,000 documents.

5 Conclusions

This was the first participation of the IMS group to the Total Recall track. Our main goal was to begin a set of experiments that will compare different strategies to the problem of high recall classification, from automatic approaches to gamification ones. We submitted four runs: two variations of the BM25 model and two variations of the two-dimensional probabilistic model, one with a simple three-step strategy to find the best decision line and another more complex that tried to mimic a human with small tweaks on the decision line. The results showed that the simpler the better in both cases: the best BM25 approach was the one with a classical smoothing approach, while the best automatic run was the simple three-step strategy. For the automatic runs, we could clearly find four different types of behaviour of the system, from the one with good performance and correct early stopping to the one with bad initial setting and continuous negative feedback that could not retrieve more than a few relevant documents. Results are encouraging giving the fact that the two-dimensional approach is indeed an active learning approach where users can, and should, suggest the

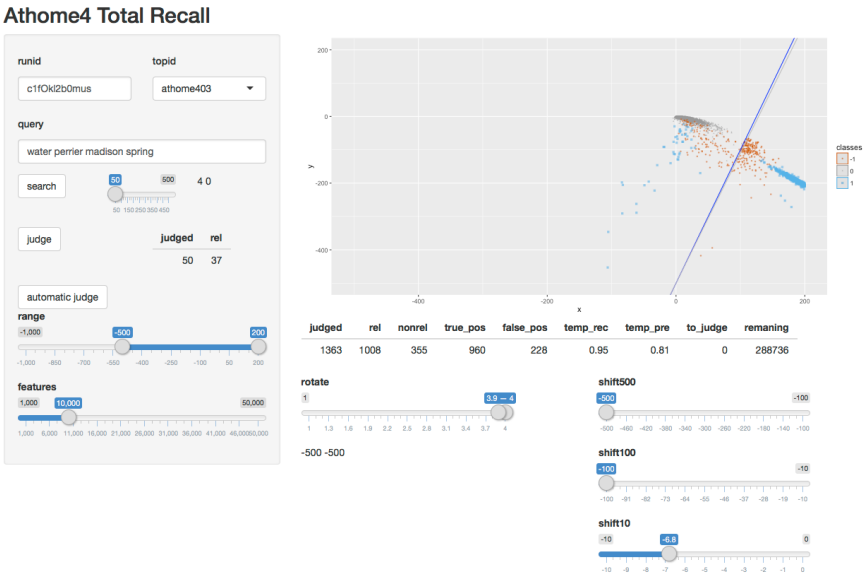
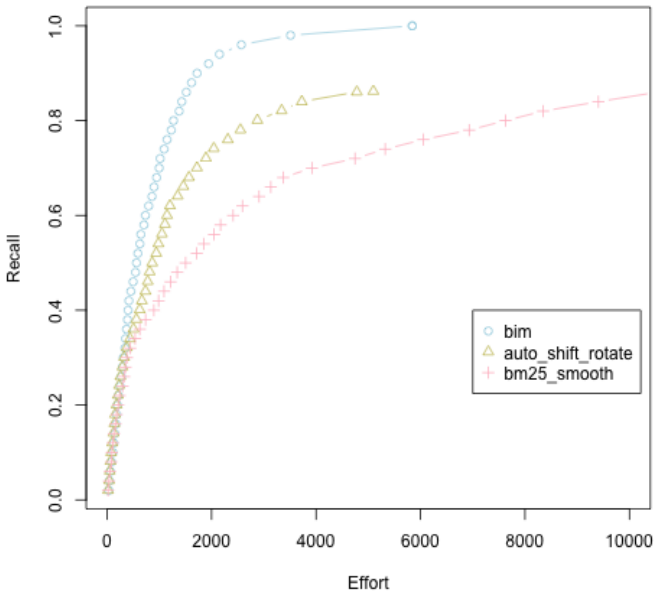


Fig. 2: Layout of the actual interactive system used for a manual check on topic 'athome403'.

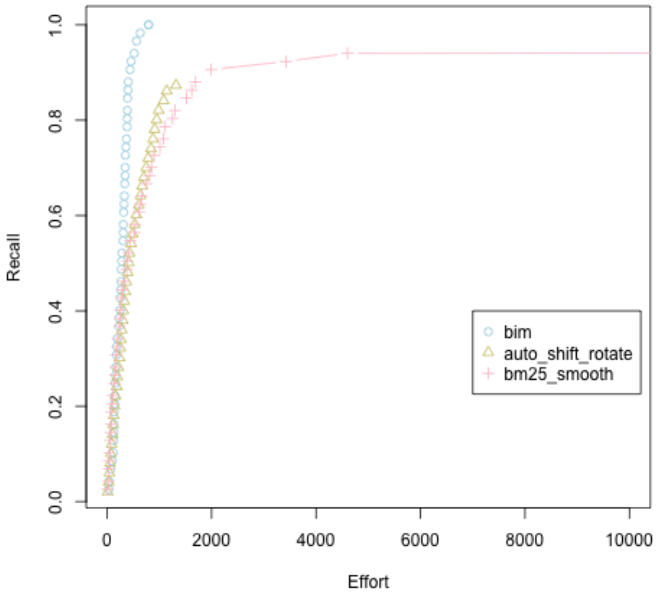
direction of the decision line with small adjustments. Future work will involve users that not only adjust the decision line but they also provide a reformulation of the query to broaden the range of relevant documents that are fed to the system.

References

1. Milan Bouchet-Valat. *SnowballC: Snowball stemmers based on the C libstemmer UTF-8 library*, 2014. R package version 0.5.1.
2. Gordon V. Cormack and Mona Mojdeh. Machine learning for information retrieval: TREC 2009 web, relevance feedback and legal tracks. In *Proceedings of The Eighteenth Text REtrieval Conference, TREC 2009, Gaithersburg, Maryland, USA, November 17-20, 2009*, 2009.
3. Giorgio Maria Di Nunzio. A New Decision to Take for Cost-Sensitive Naïve Bayes Classifiers. *Information Processing & Management*, 50(5):653 – 674, 2014.
4. Ingo Feinerer, Kurt Hornik, and David Meyer. Text mining infrastructure in r. *Journal of Statistical Software*, 25(1):1–54, 2008.
5. Giorgio Maria Di Nunzio. Shiny on your crazy diagonal. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*, pages 1031–1032, 2015.
6. Giorgio Maria Di Nunzio, Maria Maistro, and Daniel Zilio. Gamification for machine learning: The classification game. In *Proceedings of the Third International Workshop on Gamification for Information Retrieval co-located with 39th International*

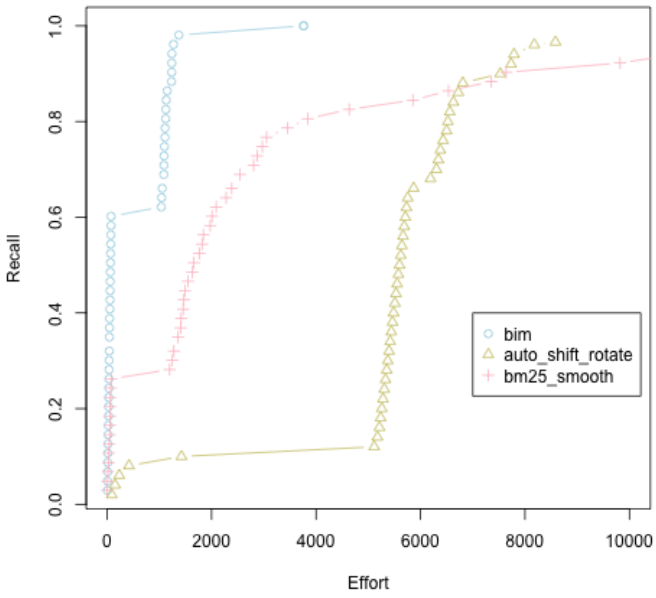


(a) Topic 402

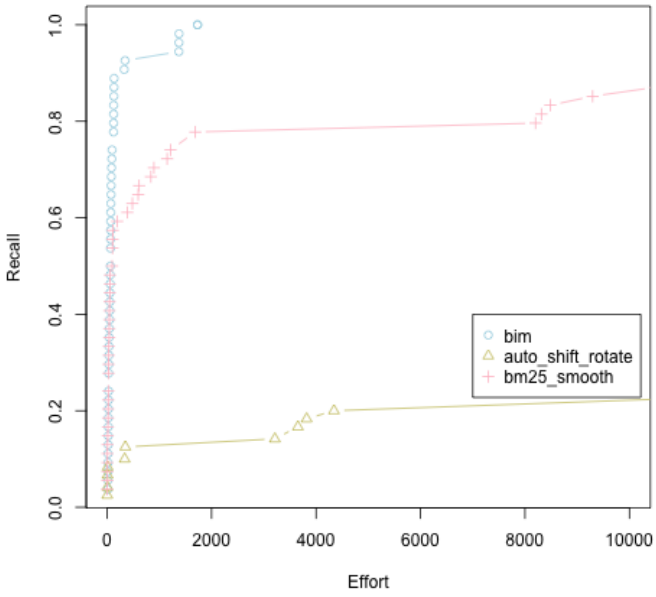


(b) Topic 424

Fig. 3: Recall - Effort examples



(a) Topic 403



(b) Topic 426

Fig. 4: Recall - Effort examples

ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2016), Pisa, Italy, July 21, 2016., pages 45–52, 2016.
7. Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, April 2009.
8. Rita Singh and Bhiksha Raj. Classification in likelihood spaces. *Technometrics*, 46(3):318–329, 2004.

A Appendix: Settings and Stopping Strategies

The four runs we submitted had four parameters in common: $\alpha^{\mathcal{R}}$, $\alpha^{\mathcal{NR}}$, $\beta^{\mathcal{R}}$, $\beta^{\mathcal{NR}}$. The number of terms was fixed to 50,000; these terms were the top 50,000 of the list of terms ordered in decreasing order of the difference $\theta_i^{\mathcal{R}} - \theta_i^{\mathcal{NR}}$. In addition, we choose different approaches for the amount of documents that had to be sent to for relevance feedback at each round.

A.1 BM25 Baseline

For the two BM25 runs, we choose different values for the four parameters in order to compare an aggressive smoothing to a classic one. We chose the same strategy for the number of documents that had to be sent for relevance feedback. In particular:

- if feedback round < 100, then top 50 (a total of 5,000 documents),
- if feedback round < 200, then top 100 (a total of 10,000 documents),
- if feedback round < 270, then top 500 (a total of 35,000 documents),
- else top 5,000 (remaining documents).

At the end of each round, the information about relevant documents is plugged in Equation 3 and 4. We did not "call the shot" on purpose.

baseline_bm25 The settings of this run were:

- $\alpha^R = 0.01$,
- $\beta^R = 5$,
- $\alpha^{NR} = 0.001$,
- $\beta^{NR} = 50$.

baseline_bm25_smoothing The settings of this run were:

- $\alpha^R = 1$,
- $\beta^R = 1$,
- $\alpha^{NR} = 0.1$,
- $\beta^{NR} = 1$.

A.2 Two-dimensional model

For the two-dimensional model, we decided to use the same 'mild' smoothing parameters of the second run of the BM25 and use two different approaches for finding the best subset of documents to assess. In both cases, the main idea is to start with a plain BM25 to retrieve at most the top k documents or stop this phase earlier if k' relevant documents are found before reaching the k-th document. Then, we start to 'sweep' the two-dimensional space with a decision line that starts with a very low (in theory minus infinity) intercept and a slope greater than one in order to find the most relevant documents (in a probabilistic sense). We stop the search when the recall on the judged documents is above a certain threshold (0.999 in our experiments).

automatic_shift_rotation We need some intial rounds of feedback to gather some relevant information. We used a BM25 without explicit relevance feedback (probabilities are not updated when relevant documents are found in these rounds) in the following way:

- select the top 10 documents and ask for relevance assessment,
- if 10 relevant documents are found or 100 documents are assessed, stop this initial search,
- otherwise repeat with the next 10 documents.

Then, the actual two-dimensional approach begins:

1. Find the coordinates of the two-dimensional space,
2. Find the interpolating line of the relevant documents and set this line as the initial decision line,
3. Find the un-judged documents that fall below the decision line,
4. While there are still documents to assess
   - call relevance assesments,
   - update estimates $\theta_i^{\mathcal{R}}$ and $\theta_i^{\mathcal{NR}}$,
   - recompute coordinates,
   - recompute the interpolating line of the relevant documents,
   - find the un-judged documents that fall below the decision line.
5. If recall is less than 0.999 and there are no documents to judge below the decision line
   - increase the intercept or rotate the decision line (explained in the following paragraph).
6. If there are documents to judge repeat from 4 otherwise stop.

During phase 5, we shift or rotate the line in the following way:

- First round (only shift): if the shift is less than -10, increase by 2, otherwise increase by 0.5 (smaller increments). Stop when shift = -0.5.
- Second round (only rotation): decrease by 0.01. Stop when rotation = 1.5.
- Third round (only shift): increase shift by 0.1 until shift = -3.

automatic_shift_rotation_exp Even in this case, we need some initial set of relevant documents. The difference with the previous run is that BM25 is updated with relevance feedback information at each of the following rounds:

- select the top 10 documents and ask for relevance assessment,
- update θ estimates with relevant information,
- if 20 relevant documents are found or 100 documents are assessed stop searching,
- otherwise repeat with next 10 documents.

Then the two-dimensional approach begins:

1. find the coordinates of the two-dimensional space,
2. find the interpolating line of the relevant documents and set this line as the decision line,
3. find the un-judged documents that fall below the decision line,
4. while there are still documents to assess:
   - call relevance assessments,
   - update estimates $\theta_i^{\mathcal{R}}$ and $\theta_i^{\mathcal{NR}}$,
   - recompute coordinates,
   - recompute the interpolating line of the relevant documents,
   - compute the interpolating line of non-relevant documents,
   - find the un-judged documents that fall below the decision line,
5. while recall is less than 0.999 and there are no documents to judge:
   - increase the intercept or rotate the decision line (explained later).
6. if there are still documents to judge repeat from 4 otherwise stop.

About phase 5, the decision to shift or rotate the line:

- First round (only shift), repeat until the intercept of the decision line is greater or equal to the intercept of the non-relevant interpolating line:
  - If shift > -10, increase shift by 0.5,
  - else if shift > -50, increase shift by 1,
  - else if shit > -100, increase shift by 2,
  - else if shift > -200, increase shift by 5,
  - otherwise increase shift by 10.
- Second round (only rotation): decrease rotation by a fixed threshold. Stop when rotation equals the slope of the interpolating line of the relevant documents (adjusted if necessary).
- Third round (only shift): increase shift by a fixed threshold until shift equals the intercept of the non-relevant interpolating line (adjusted if necessary).
- Stop when recall > 0.999.