TUW AT THE FIRST TOTAL RECALL TRACK

MIHAI LUPU

Abstract. For the first participation in the TREC Total Recall track, we set out to try some basic changes to the baseline provided by the organisers. Namely, the weighting scheme, the use of stopwords, and the number of learners that contribute to the decision of which documents to ask the virtual assessor to review. We observed that the baseline was extremely strong and none of the runs significantly and consistently outperformed it.

1. Introduction

As the organizers point out, the focus of the Total Recall Track is to evaluate methods to achieve very high recall, including methods that include a human assessor in the loop [6].

We submitted six automated runs for the small *At home* task and provided scripts for the sandbox evaluation. The `athome1` data contains 290099 files grouped in 115 folders, with 333 files in the smallest, 16850 in the largest, a mean of 2522.6 files per folder and a median of 2228. Table 1 shows the 10 topics used for the `athome1` part.

Two other collections were tested in the sandbox environment: the MIMIC II clinical dataset[1] (`C`) and the Kaine Email Collection[2] (`Kaine`) indexed each 31174 and, respectively, 401953 documents.

[1] `https://physionet.org/mimic2/mimic2_clinical_overview.shtml`

[2] `http://www.virginiamemory.com/collections/kaine/`

Table 1. `athome1` topics

Topic	**Information Need**
athome100	School and Preschool Funding
athome101	Judicial Selection
athome102	Capital Punishment
athome103	Manatee Protection
athome104	New medical schools
athome105	Affirmative Action
athome106	Terri Schiavo
athome107	Tort Reform
athome108	Manatee County
athome109	Scarlet Letter Law

2. Approach

For this year's participation, we have only modified the provided Baseline Model Implementation (BMI) [2] to test some very simple changes to the method. Namely, we looked at the use of stopwords (runs indicated by the presence of an "S" in their name), the use of a different term weighting scheme—a recently introduced adaptation of BM25 [3] that does not need optimizing for the b parameter, and a modified voting of 6 learners instead of using only 1. We discuss the weighting in the following subsection. The learner voting mechanism appears in Section 2.2.

The common pre-processing are fundamentally the same as those of the BMI. We only counted additional data necessary in the adapted BM25.

tokenisation: tokens are identified by first splitting on non-alphanumeric characters and then removing all strings thus obtained that contain at least one digit. All tokens of length 1 were ignored.

casing: all tokens were lowercased

stopwords: only for runs marked by "S", tokens appeared in the list in the Appendix A were ignored.

2.1. **Term weighting.**

The BMI used the basic tf.idf weighting scheme, as given by:

$$weight_T(t,d) = (1 + log(tf_{t,d})) * log(N/df_t); \tag{1}$$

where t is a term, d a document, $tf_{t,d}$ the term frequency, df_t the document frequency, and N is the number of documents in the collection.

For our weighting, we used an observation recently by Lipani et al. [3], that using the average term frequency in a document and the mean average term frequency over the collection, we can define for BM25 a b parameter that is collection specific. This would be particularly useful here, since it would save us some training effort. The used weight, marked by "B" in the run names, is given by:

$$weight_B(t,d) = \frac{tf_{t,d}}{\left(\frac{1}{mavgtf}\frac{avgtf_d}{mavgtf} + (1 - \frac{1}{mavgtf})\frac{L_d}{avgdl}\right)k_1 + tf_{t,d}} \frac{N - df_t + 0.5}{df_t + 0.5} \tag{2}$$

where, apart from the variables already presented for Eq. 1, k_1 is the usual BM25 parameter controlling tf normalization, $avgtf_d$ and L_d are the average term frequency in document d and, respectively, its length. Finally, $mavgtf$ and $avgdl$ are the mean average term frequency and the average document length, calculated over all documents. Throughout the experiments, k_1 was maintained at its "standard" value of 1.2. We note that there exists previous work that removes the need for optimizing on k_1 [4], which could be applied here. At this time, it remains for future work.

2.2. **Learner voting.**

The BMI uses the Sofia ML suite for incremental machine learning algorithms [7]. In particular it uses `logreg-pegasos`, i.e. Logistic Regression with Pegasos updates, optimizing over ROC area, with 200k iterations, dimensionality 1.1mil, and $\lambda = 10^{-4}$. These were all parameters established by the track organisers, and while we fiddled at times with them, we found no compelling reason to change them.

The only change we made was at a higher level. The Sofia ML library provides 5 more ML algorithms. The following list is quoted directly from the manual, we refer the reader to the website[3] and D. Sculley's publications [7] for further details.

pegasos: Use the Pegasos SVM learning algorithm. `--lambda` sets the regularization parameter, with values closer to zero giving less regularization. Note that Pegasos enforces a hard constraint that the model weight vector must lie within an L2 ball of radius at most 1/sqrt(lambda). Also relies on `--eta_type`

sgd-svm: Use the SGD-SVM learning algorithm. –lambda sets the regularization parameter, with values closer to zero giving less regularization. Also relies on `--eta_type` passive-aggressive Use the Passive Aggressive Perceptron learning algorithm. `--passive-aggressive-c` sets the largest step size to be taken on any update step; this operates as a capacity term with values closer to zero encouraging simpler models. `--passive-aggressive-lambda` will force the model weight vector to lie within an L2 ball of radius 1/sqrt(passive-aggressive-lambda)

margin-perceptron: Use the Perceptron with Margins algorithm. Sets the update margin with `--perceptron-margin-size`. When set to 0, this is exactly equivalent to the classical Perceptron by Rosenblatt. When set to 1, this is equivalent to optimizing SVM hinge-loss without regularization. Increasing values may give additional tolerance to noise. Also relies on `--eta_type`.

romma: Use the ROMMA algorithm. No parameters to set.

logreg-pegasos: Use Logistic Regression with Pegasos updates; we optimize logistic loss and enforce Pegasos-style regularization and constraints, with `--lambda` being the regularization parameter. Also relies on `--eta_type`. Note that the classification values provided by this method regression are logodds, and can be converted to probabilities using: exp(p) / (1 + exp(p)).

The runs using all six learners (denoted by "6" in their name), during each iteration of theBMI take first all the documents on which all learners agree, then those on which 5 agree, then those on which 4 agree. After that, they complete the set with the documents proposed by `logreg-pegasos` but are not yet in the set to be sent for evaluation. This was especially necessary in the first few iterations where extremely little agreement was found between learning methods.

3. Results

For each recall value, we performed an ANOVA to test the omnibus hypothesis that all the runs are equal by Precision. In most cases, and particularly for high recall values, this hypothesis could not be rejected and therefore we cannot say that any of the runs are actually different from the baseline. Where the hypothesis was rejected, we performed pairwise tests and we report those. All tests were performed using Carterette's R implementation [1].

The sandbox `C` collection (Mimic 2 Clinical Decision Dataset) presents a strange behaviour in the sense that it never reaches recall 1. From our own data, we see that all runs

[3] `https://github.com/huitseeker/sofia-ml`

run	athome1				C				Kaine			
	recall 0.95		recall 1.00		recall 0.95		recall 1.00†		recall 0.95		recall 1.00	
	prec	f1	prec	f1	prec	f1	prec	f1	prec	f1	prec	f1
bmi	0.51	0.60	0.05	0.09	0.46	0.59	0.25	0.37	0.57	0.70	0.21	0.32
1NB	0.52	0.60	0.06	0.10	0.44	0.58			0.55	0.69		0.31
1SB	0.52	0.60	0.05	0.09	0.45	0.58			0.55	0.69		0.31
1ST	0.50	0.60	0.04	0.08	0.45	0.58			0.57	0.70		0.32
6NB	0.52	0.61	0.05	0.09	0.44	0.57			0.56	0.69		0.31
6SB	0.47	0.56	0.05	0.08	0.44	0.58			0.57	0.70		0.31
6ST	0.49	0.58	0.04	0.08	0.45	0.58			0.58	0.71		0.32

† for all topics, after rounding up to nearest 0.01 (topic C11 was rounded up from 0.9945)

and all topics reach a maximum of exactly 31174, from which our assumption that the dataset contains these many documents. However, the User Guide of the dataset[4] states that the April 2011 release (version 2.6) contains around 33k patients. Apparently, the difference consists of documents without any text. Multiplying the reported recall with the known number of relevant documents per topic, we observe that, while exploring the maximum set of 31174 indexed documents, we are missing, on average, 22.79 documents per topic. For most topics this is above 0.995 recall and therefore would be 1.00 when rounding up to the nearest cent (10^{-2}). For topic `C11` however, the total number of relevant documents is only 180, and one missing document results in a recall of 0.9945, which rounds up to 0.99. For plotting, we forced this to 1.00 as well, to maintain visibility.

Another data alteration we do for consistency is to assign recall 1.05 to the effort and precision results reported when using the entire dataset. Therefore, when we talk about recall 1.00, we refer to the first time this recall value was obtained. Otherwise, we will talk about recall on the dataset (which is generally 1.00 except for topic `C11` mentioned above).

Figures 1 and 2 show the precision recall curves for the three test collections and for each topic, respectively. For `athome1`, the curves are statistically indistinguishable, except for points at recall 30% and 50%. For `C`, the curves are completely indistinguishable, and as for `Kaine`, the Webis is significantly lower than the other runs.

By Precision-Recall curve, probably the most interesting run is `athome109 (Scarlet Letter Law)`, as its precision increases with recall almost up to recall 1. The topic had 506 relevant documents in the collection. The information need is, presumably, any information about laws whose main or side-effect is a public shaming of individuals, but it may be also referring only to a specific law passed and then repealed in Florida. A quick grep on the collection shows that there are 22 documents actually containing the two words "scarlet" and "letter" separated by 3 characters or less. All of these documents contain also the term "Jeb" (case sensitive, representing the first name of former Florida governor, Jeb Bush). Individually, "scarlet" (ignoring case and surrounded by non-alphanumeric characters) appears in 70 documents, 66 of which also contain the term "Jeb". The 4 other documents refer to people named Scarlet (a more common spelling of this name is with a

[4] `https://physionet.org/mimic2/UserGuide/node15.html`

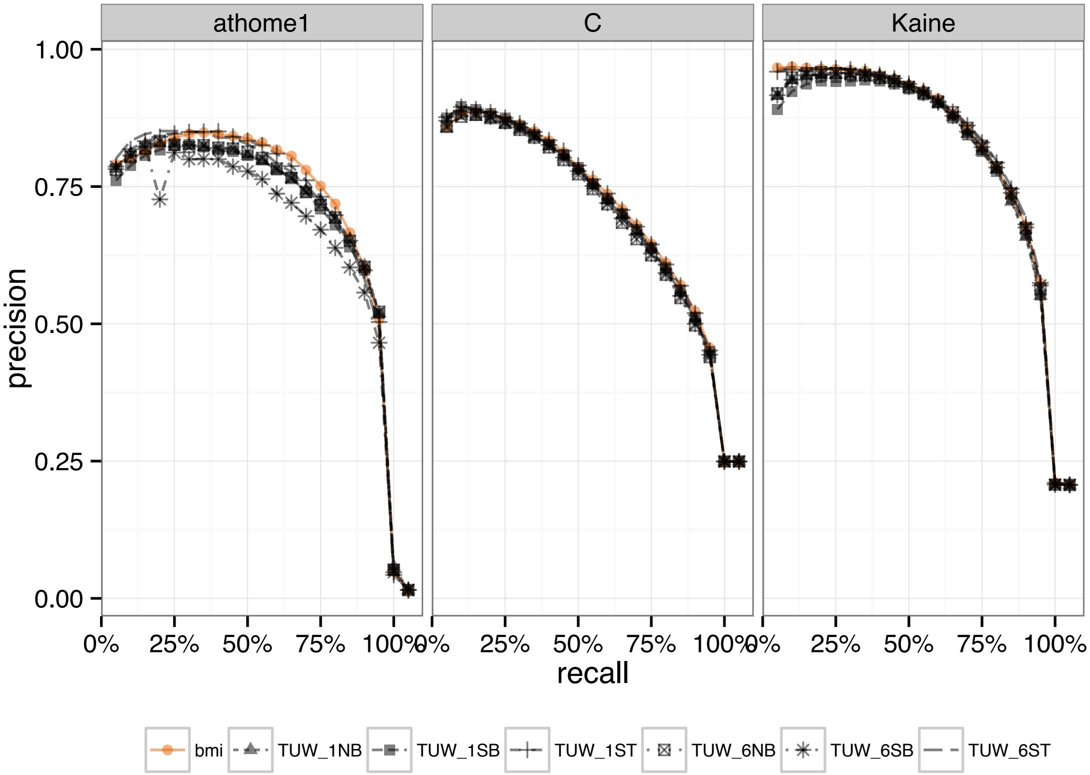

Figure 1. Average Precision-Recall curve for each collection. Points beyond 100% recall represent precision after the entire dataset was evaluated.

double ending consonant). From these 70 to the total of 506 the system has to figure it out using only "letter" and "law", two relatively common terms.

Figures 3 and 4 show the recall versus effort, as calculated by the organizers. Figures 3 shows the average over all topics, by collection, run, and coefficients a and b. Figure 4 shows details for each topic, for a fixed $b = 0$.

4. Conclusion

Our submissions this year did not improve upon the provided baseline. This appears to have been the general observation of this year's track: *"Several manual and automatic participant efforts achieve higher recall with less effort than the baseline on some topics, but none consistently improves on the baseline"* [5]. In our case, the use of stopwords appears to be counter productive, even though we used very few of them. The modified weighting scheme showed insignificant improvements on `athome1`. The use of multiple

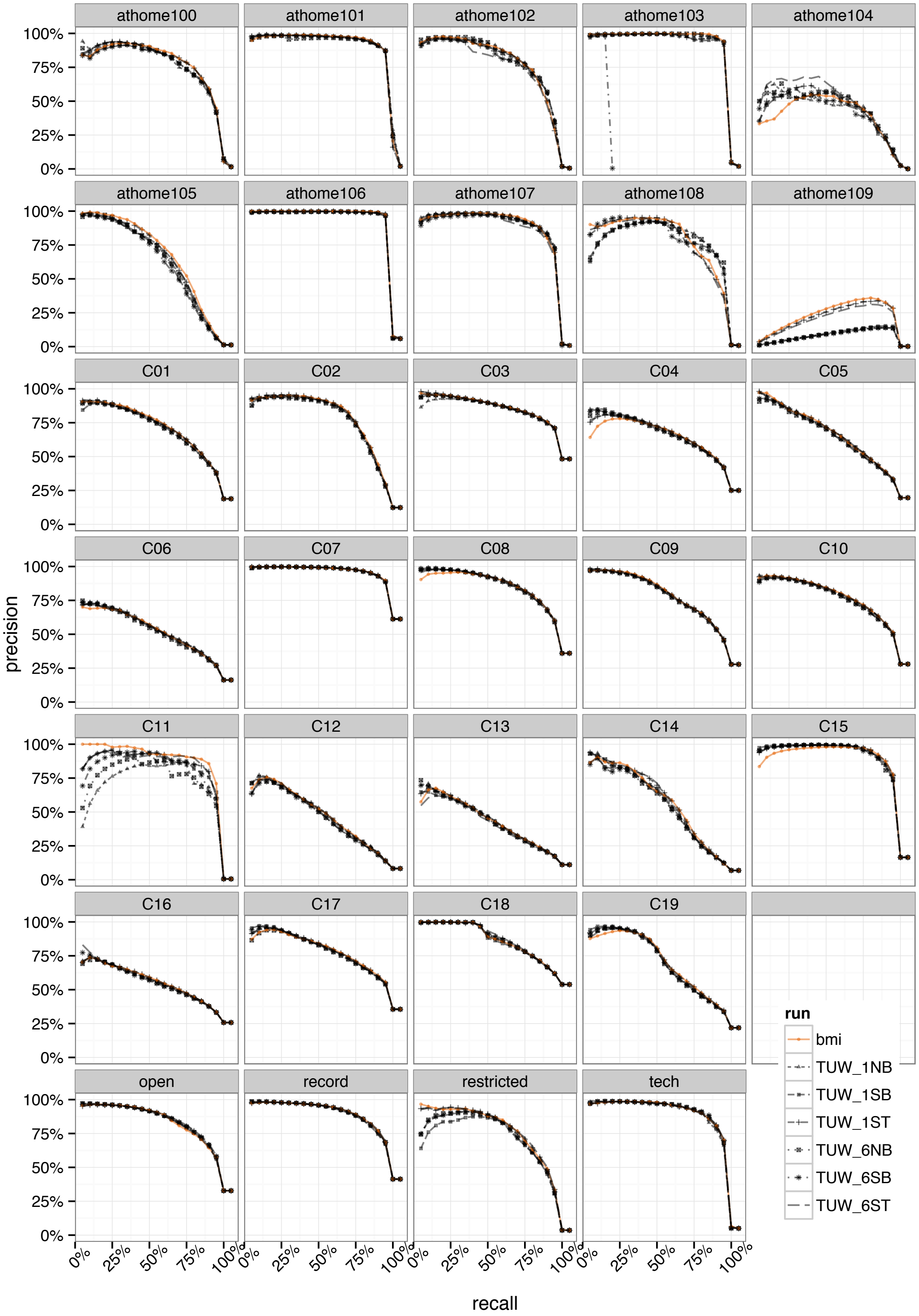

FIGURE 2. Precision-Recall curve for each topic and all runs. Points beyond 100% recall represent precision after the entire dataset was evaluated.

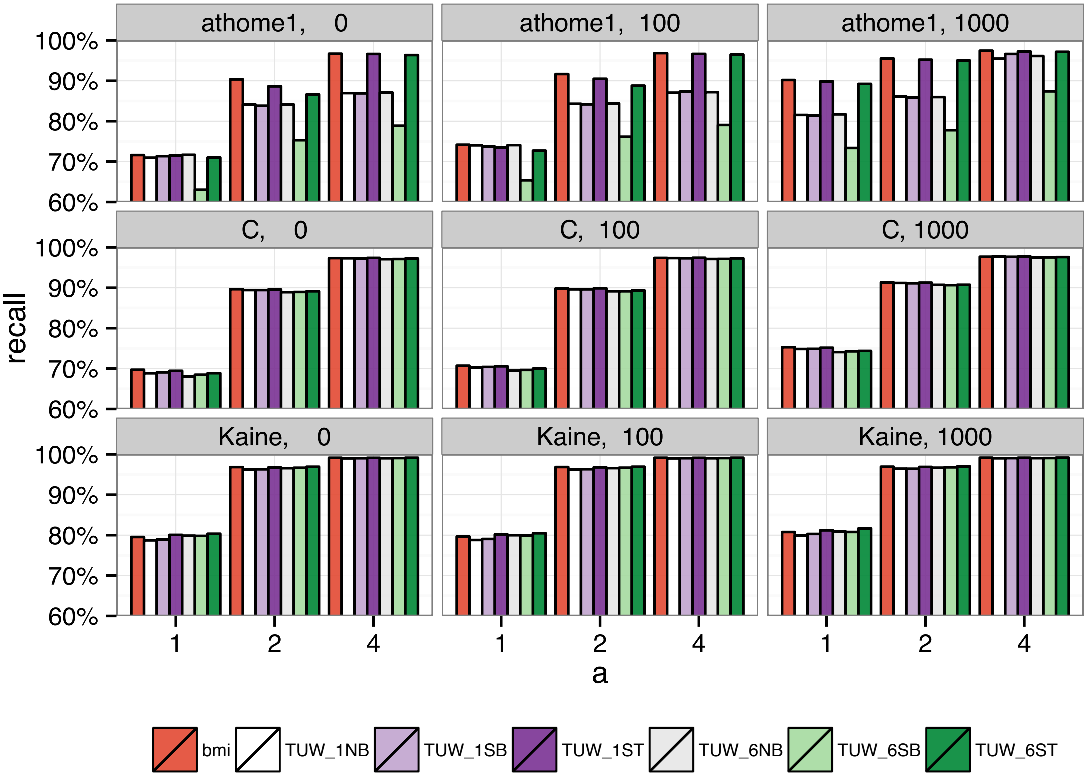

FIGURE 3. Average Effort. The b coefficient is in the title of each sub-plot

learners increased processing time significantly without having any (positive) effect on effectiveness.

APPENDIX A. STOP WORDS LIST

a, about, above, after, again, against, all, am, an, and, any, are, aren't, as, at, be, because, been, before, being, below, between, both, but, by, can't, cannot, could, couldn't, did, didn't, do, does, doesn't, doing, don't, down, during, each, few, for, from, further, had, hadn't, has, hasn't, have, haven't, having, he, he'd, he'll, he's, her, here, here's, hers, herself, him, himself, his, how, how's, i, i'd, i'll, i'm, i've, if, in, into, is, isn't, it, it's, its, itself, let's, me, more, most, mustn't, my, myself, no, nor, not, of, off, on, once, only, or, other, ought, our, ours, ourselves, out, over, own, same, shan't, she, she'd, she'll, she's, should, shouldn't, so, some, such, than, that, that's, the, their, theirs, them, themselves, then, there, there's, these,

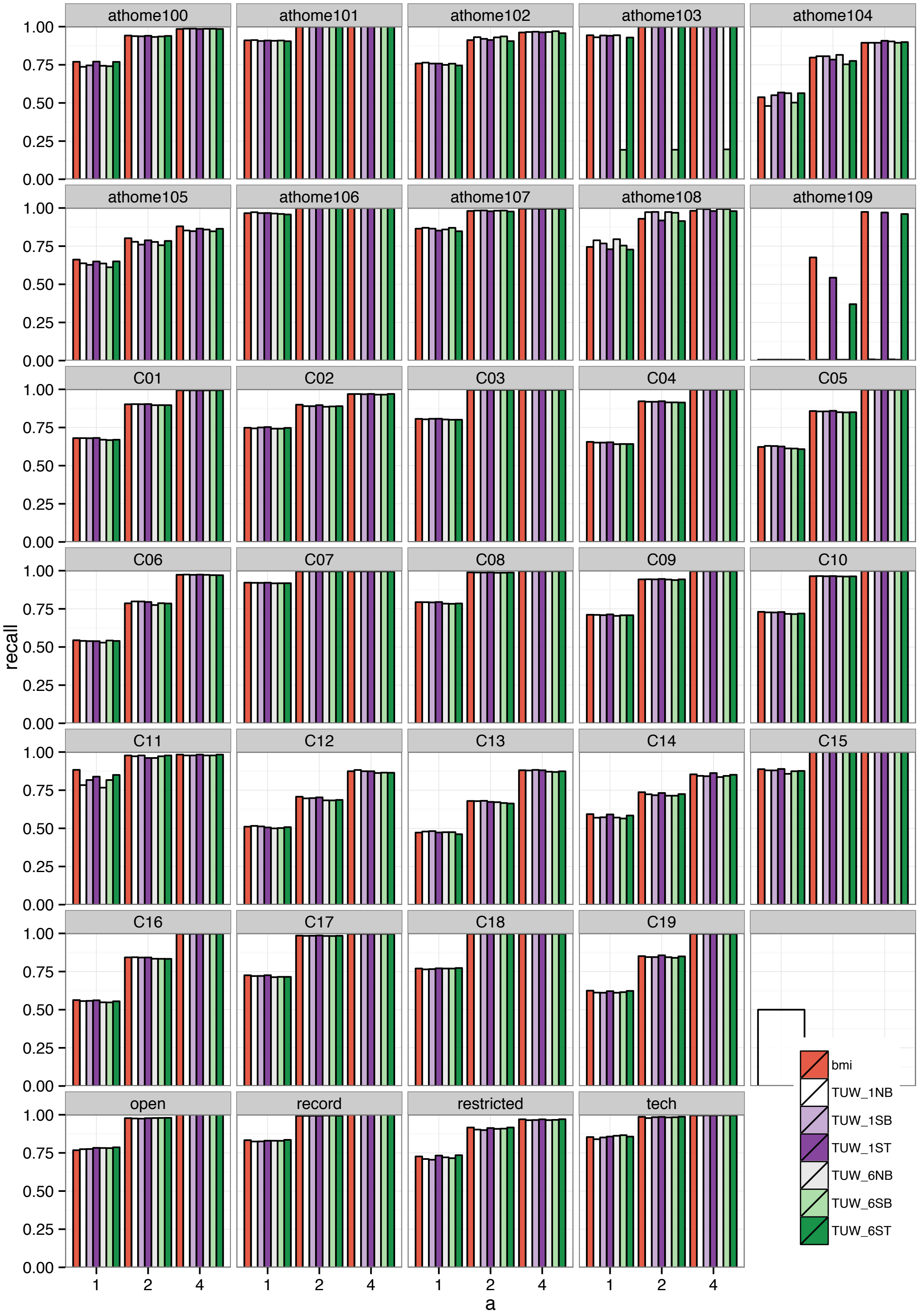

Figure 4. Effort for each topic for b=0

they, they'd, they'll, they're, they've, this, those, through, to, too, under, until, up, very, was, wasn't, we, we'd, we'll, we're, we've, were, weren't, what, what's, when, when's, where, where's, which, while, who, who's, whom, why, why's, with, won't, would, wouldn't, you, you'd, you'll, you're, you've, your, yours, yourself, yourselves

References

[1] Benjamin A. Carterette. Multiple Testing in Statistical Analysis of Systems-based Information Retrieval Experiments. *ACM Trans. Inf. Syst.*, 30(1), 2012.

[2] Gordon V. Cormack and Maura R. Grossman. Evaluation of Machine-learning Protocols for Technology-assisted Review in Electronic Discovery. In *Proc of SIGIR*, 2014.

[3] Aldo Lipani, Mihai Lupu, Allan Hanbury, and Akiko Aizawa. Verboseness Fission for BM25 Document Length Normalization. In *Proc. of ICTIR*, 2015.

[4] Yuanhua Lv and ChengXiang Zhai. A log-logistic model-based interpretation of tf normalization of bm25. In *Proceedings of the 34th European Conference on Advances in Information Retrieval*, ECIR'12, pages 244–255, Berlin, Heidelberg, 2012. Springer-Verlag.

[5] Adam Roegiest, Gordon V. Cormack, Charles L.A. Clarke, and Maura R. Grossman. Notebook draft trec 2015 total recall track overview. In *Proc. of TREC*, 2015.

[6] Adam Roegist, Gordon Cormack, Maura Grossman, and Charles Clarke. Total recall track overview. In *Proc. of TREC*, 2015.

[7] D. Sculley. Combined Regression and Ranking. In *Proc. of SIGKDD*, 2010.