The Simplest Thing That Can Possibly Work: Pseudo-Relevance Feedback Using Text Classification

Jimmy Lin

David R. Cheriton School of Computer Science University of Waterloo jimmylin@uwaterloo.ca

ABSTRACT

Motivated by recent commentary that has questioned today's pursuit of ever-more complex models and mathematical formalisms in applied machine learning and whether meaningful empirical progress is actually being made, this paper tries to tackle the decadesold problem of pseudo-relevance feedback with "the simplest thing that can possibly work". I present a technique based on training a document relevance classifier for each information need using pseudo-labels from an initial ranked list and then applying the classifier to rerank the retrieved documents. Experiments demonstrate significant improvements across a number of newswire collections, with initial rankings supplied by "bag of words" BM25 as well as from a well-tuned query expansion model. While this simple technique draws elements from several well-known threads in the literature, to my knowledge this exact combination has not previously been proposed and evaluated.

1 INTRODUCTION

The breakneck pace of advances in machine learning, particularly deep learning applied to vision, speech, and text processing tasks, has recently prompted a number of researchers to urge caution and the need for self reflection. Sculley et al. [22] and Lipton and Steinhardt [14] represent two recent commentary along these lines. The trend of increasingly complex models with poor ablation studies to attribute gains, coupled with the use of mathematics to obfuscate or to impress reviewers, has put empirical research on shaky footing. In a similar vein, I've recently expressed skepticism about whether neural ranking models actually improve over existing models, at least absent large amounts of behavioral training data [13].

This paper tackles the decades-old problem of pseudo-relevance feedback. I am guided by the advice of Ward Cunningham (inventor of the wiki), which is to ask yourself, "What's the simplest thing that could possibly work?" Here's the answer I came up with:

Given a standard $ad\ hoc$ retrieval setup, a ranking model produces a ranked list H with respect to an information need represented by query Q; this is referred to as the base run. Following the general setup of pseudo-relevance feedback, let's assume that the first r hits are relevant, i.e., H[:r] in Python's array slice notation. Let's further assume that the last n hits of the ranked list are not relevant, i.e., H[-n:] in Python's array slice notation. These r+n documents with their pseudo labels are then used to train a text classifier over the tf-idf representations of the document terms. This paper explores logistic regression, SVMs, and a simple ensemble. The trained classifier is then applied to score all documents in the base run: a new document score comprised of a linear interpolation

between the initial retrieval score and the classifier score is then used to create the final ranked list (for evaluation).

Experiments on four newswire collections show that this simple technique yields significant increases in effectiveness over a base run from "bag of words" BM25 as well as a base run that already exploits pseudo-relevance feedback via RM3. The latter result suggests that the proposed technique provides additive improvements on top of a strong baseline.

2 PRIOR WORK

The obvious retort to this "simplest thing that can possibly work" is that, even if it works, it can't possibly be novel! While the literature does contain reports of similar ideas, this exact combination to my knowledge has not been tried before.

The idea of treating document ranking as a binary classification problem, to distinguish relevant from non-relevant documents, has a long history, dating back to the binary independence retrieval (BIR) model of Robertson and Spark Jones [19]. In the modern parlance of learning to rank, this is commonly known as a *pointwise model* [15]. As an early example, Nallapati [16] used logistic regression and SVMs for document ranking with features based on *tf*, *idf*, and statistics derived from their combination.

Relevance feedback in IR systems dates back to the 1960s [20] and the idea of using pseudo labels for relevance feedback dates back to at least the late 1970s [7]; a nice historical overview is offered by Ruthven and Lalmas [21]. According to Buckley [3], early attempts at pseudo-relevance feedback were unsuccessful because of small collection sizes, and it wasn't until around TREC-2 that researchers demonstrated positive results. Today, the effectiveness of pseudo-relevance feedback is well established in the literature.

In most setups, including the popular RM3 approach widely used today [1], the first k hits of an initial ranked list are assumed to be relevant. Some analysis (varies by approach) is applied to these pseudo-relevant documents to generate an expanded query that is then used to produce a final ranked list. Typically, this class of methods only exploits pseudo-positive labels. Far less popular, the use of pseudo-negative labels dates back to at least 2003 [24] (the earliest reference that I could find), albeit in the context of multimedia retrieval. A more recent example is the use of pseudo-negative labels by Raman et al. [18] to extract better query expansion terms. Cormack et al. [6] used pseudo-negative labels to train spam classifiers in a distantly-supervised manner. Note that while there is literature on how to select good expansion terms using *supervised* machine learning techniques [4], my approach is completely *unsupervised*.

There is a thread of research putting together relevance feedback with text classification: Cormack and Mojdeh [5] applied logistic regression classifiers for relevance feedback in TREC 2009. Grossman

and Cormack [9] later proposed a variant where classifiers were trained on relevance judgments on a different collection for the same information need; this work was successfully reproduced by Yu et al. [27]. Xu and Akella [23] described an active relevance feedback approach using logistic regression. All these cases, however, took advantage of human relevance judgments and did not consider pseudo judgments. I argue that this distinction, while perhaps obvious in retrospective, is quite important—historically, the idea of using pseudo labels came at least a decade after the introduction of relevance feedback, and its empirical value wasn't demonstrated until many years after that (see above). Yan et al. [24] described a technique quite similar to what I propose here, expect applied to multimedia retrieval.

Pulling all the pertinent characteristics together—use of both positive and negative pseudo-labels to train text classifiers in a pseudo-relevance feedback setup for *ad hoc* retrieval—I assert that this paper is the first to propose such a technique and present experimental results on a number of modern test collections.

3 EXPERIMENTAL SETUP

All experiments were conducted using Anserini [25], an opensource IR toolkit built on top of Lucene. Anserini provides convenient tools to dump out raw tf-idf document vectors for arbitrary documents, which was used to extract feature vectors for the top rand bottom n documents from the base run for each topic. As part of preprocessing, all feature vectors were converted to unit vectors by L_2 normalization.

For each topic, a training set was created from the documents corresponding to the positive and negative pseudo labels, as described in the introduction. Feature vectors were then fed to the Python package <code>scikit-learn[17]</code> (v0.20.1). I tried three different models: logistic regression (LR), SVMs with a linear kernel, and an ensemble of the two using simple score averaging. In each case, the trained classifier for each topic was then applied to all documents in the ranked list for that topic from the base run. Documents in the base run were reranked using a linear interpolation between retrieval and classifier scores.

Experiments were conducted on four newswire collections:

- Robust04: TREC Disks 4 & 5 (excluding Congressional Record) with topics and judgments from the TREC 2004 Robust Track.
- Robust05: The AQUAINT Corpus, with topics and judgments from the TREC 2005 Robust Track.
- Core17: The New York Times Corpus, with topics and judgments from the TREC 2017 Common Core Track.
- Core18: The Washington Post Corpus, with topics and judgments from the TREC 2018 Common Core Track.

The primary test collection was Robust04, to take advantage of baselines and comparisons I've previously established [13]. The other newswire collections provide some indication of the generality of the technique, at least for documents of the same genre. Note that web collections were not considered because most were created with shallow pools, which make them inappropriate for studies on query expansion due to prodigious amounts of missing judgments; see Yang and Lin [26] for an example analysis.

	Condition	AP	<i>p</i> -value				
	Baseline BM25 (5-fold)						
1	BM25	0.2531					
2	BM25 + LR	0.2734	2.67×10^{-7}				
3	BM25 + SVM	0.2685	1.29×10^{-8}				
4	BM25 + ensemble	0.2724	2.71×10^{-8}				
	Default RM3 parameters (5-fold)						
5	BM25 + RM3	0.2903					
6	BM25 + RM3 + LR	0.3002	0.0001225				
7	BM25 + RM3 + SVM	0.2986	2.56×10^{-5}				
8	BM25 + RM3 + ensemble	0.2998	1.34×10^{-5}				
	RM3 cross-validation: 2-fold from Paper 1						
9	BM25 + RM3	0.2987					
10	BM25 + RM3 + LR	0.3035	0.0241963				
11	BM25 + RM3 + SVM	0.3031	0.0015774				
12	BM25 + RM3 + ensemble	0.3023	0.0112373				
	RM3 cross-validation: 5-fold from Paper 2						
13	BM25 + RM3	0.3033					
14	BM25 + RM3 + LR	0.3092	0.0105262				
15	BM25 + RM3 + SVM	0.3096	1.02×10^{-5}				
16	BM25 + RM3 + ensemble	0.3082	0.0018256				
17	Paper 1	0.2720					
18	Paper 2	0.2971					
19	NPRF	0.2904					
20	Best TREC (pircRB04t3)	0.3331					

Table 1: Effectiveness of pseudo-relevance feedback using text classification on Robust04.

My proposed technique has three parameters: r, the number of pseudo-positive labels, n, the number of pseudo-negative labels, and α , the interpolation weight. Preliminary exploration showed n=100 to be a good setting (relatively insensitive) and that $r \in \{10, 20, 30\}$ seemed to be good choices. For the interpolation parameter, all values between 0.0 and 1.0 in tenth increments were tried. All parameter tuning was accomplish via cross validation.

4 RESULTS

Experimental results on Robust04 in terms of average precision at rank 1000 are shown in Table 1. In the rows, "LR", "SVM", and "ensemble" refer to different text classification models discussed in Section 3. Rows 2-4 report my technique applied to a "bag of words" BM25 run ($k_1 = 0.9, b = 0.4$). Rows 6–8 report my technique applied to RM3 (using default parameters from the open-source Indri Search Engine) on top of a BM25 base run. In both cases results are reported for parameter tuning $(r, n, and \alpha)$ using fivefold cross validation. For these and all subsequent experiments, statistical significance of metric differences was assessed using a paired two-tailed t-test. Cognizant of the dangers of multiplehypothesis testing, the right column reports the exact p-values, which allows readers to make corrections for multiple hypothesis testing as they feel appropriate. Results show that my proposed technique (all models) unequivocally improves average precision (even, for example, after applying a Bonferroni correction).

2

 $^{^{1}}$ Commit id 9548cd6, dated 01/19/2019.

		Robust05		Core17		Core18	
	Condition	AP	<i>p</i> -value	AP	<i>p</i> -value	AP	<i>p</i> -value
1	BM25	0.2031		0.1977		0.2491	
2	BM25 + LR	0.2457	0.000203	0.2318	0.002472	0.2791	0.026064
3	BM25 + SVM	0.2404	0.000720	0.2228	0.004935	0.2798	0.000309
4	BM25 + ensemble	0.2446	0.000395	0.2298	0.002677	0.2743	0.034362
5	BM25 + RM3	0.2602		0.2682		0.3147	
6	BM25 + RM3 + LR	0.2820	0.001307	0.2882	0.001146	0.3214	0.313905
7	BM25 + RM3 + SVM	0.2798	0.001533	0.2855	0.000178	0.3273	0.011007
8	BM25 + RM3 + ensemble	0.2814	0.001549	0.2880	0.000318	0.3286	0.030315
9	TREC best (automatic)	0.3096		0.2752		0.2761	-

Table 2: Effectiveness of pseudo-relevance feedback using text classification on Robust05, Core17, and Core18.

It is worth pointing out that this technique only acts as a reranker—it cannot "produce" any relevant document that was not in the base run. Thus, the improvements over BM25 come solely from bringing relevant documents into higher ranks. Since RM3 performs query expansion in a second round retrieval, it is able to retrieve more relevant documents; my technique can further improve the ranks of those documents.

In the above conditions no effort was made to tune parameters for the base runs. The next set of experiments examined whether pseudo-relevance feedback using text classification can further improve over well-tuned base runs, building on the comparisons to "Paper 1" and "Paper 2" in my recent article [13]. In Table 1, rows 9 and 13 replicate those tuned baselines, on top of which another round of reranking was applied. Results are shown in rows 10–12 for Paper 1 and rows 14–16 for Paper 2. Based on the *p*-values, the improvements would be considered significant, although not so after a Bonferroni correction, except for SVM.

It should be emphasized that not only does pseudo-relevance feedback using text classification demonstrate improvements over different base runs, but also that the reported metrics are quite high in absolute terms. For reference, the best results reported in Paper 1 and Paper 2 are copied in rows 17 and 18 for reference. Also included are results from NPRF, a recently-proposed neural approach to pseudo-relevance feedback [12] (row 19). The base runs already surpass the highest effectiveness reported in all these papers, and my technique further increases retrieval effectiveness. In other words, pseudo-relevance feedback using text classification is both better and simpler. Nevertheless, its effectiveness still falls short of pircRB04t3, the best run from TREC 2004 (row 20).

Results from the three other collections are shown in Table 2, organized in the same manner as Table 1. Reported are the applications of my technique on top of "bag of words" BM25 as well as BM25 with RM3 (both with default parameters, comparable to rows 1 and 5 in Table 1). As with the previous experiments, the parameters r,n, and α were tuned via five-fold cross-validation. I have not extended the Robust04 tuning experiments to these collections, so additional points of comparison are not available. For reference, row 9 presents the best automatic run at that particular year's TREC.

These results are consistent with the results on Robust04, providing evidence that my technique generalizes across different collections. For Robust05 and Core17, all gains appear to be statistically significant, although for Core18 only the SVM models are (despite the fact that metric increases are comparable in magnitude for logistic regression). In absolute terms, for Core17 and Core18, all results on BM25 + RM3 are higher than the best automatic TREC run submitted that year.

Across all four collections, logistic regression in most cases is slightly better than SVMs in terms of effectiveness, but the gains from logistic regression are not significant for all collections. Results further suggest that an ensemble based on simple score averaging yields no benefit over individual models.

5 DISCUSSION

Given the simplicity and effectiveness of my proposed technique for pseudo-relevance feedback using text classification, the obvious question is: Where are the gains coming from? While it may be no surprise that the technique improves a "bag of words" BM25 base run, it also improves RM3, a base run that already exploits pseudo-relevance feedback. The gains, albeit smaller, are significant (across all collections, at least for SVMs). This suggests that my technique is extracting additional relevance signal beyond what RM3 can identify.

The explanation, I believe, lies in what Diaz [8] calls score regularization in ranked retrieval, which is the idea that closely-related documents should have similar scores. This itself is a restatement of the decades-old cluster hypothesis [10], which is the observation that relevant documents tend to share similar content (i.e., cluster in document space). Thus, effectiveness gains come from breaking the independence assumption in ranking, which still holds in RM3. Another way to summarize these results is that these observed effectiveness improvements are additive with respect to RM3. This is an important finding because the question of additivity has significant bearing on the methodology of empirical research in information retrieval. Previous work [2, 11] has found that many techniques only improve over weak baselines: when applied to stronger baselines, observed gains disappear. In other words, the relevance signals that these techniques exploit are subsumed by the stronger baselines. In this context, the regularization effect potentially explains why improvements are additive: it taps a different source of signal.

3

 $^{^2{\}rm Note}$ that these results are slightly higher than those reported in the SIGIR Forum article due to improvements made after publication.

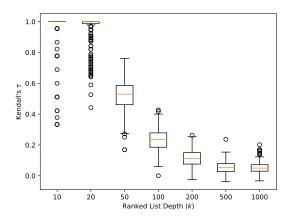


Figure 1: Results of a rank correlation analysis on Robust04, comparing BM25 vs. BM25+RM3 base runs at different k.

A closer look at the base and reranked runs reveals interesting observations. For example, compare row 13 vs. row 15 in Table 1 as a representative case. On a per-topic basis, SVMs helped 70 topics, arbitrarily defined as AP increases > 0.01; SVMs hurt 36 topics, arbitrarily defined as AP decreases > 0.01; for the remaining topics (143) SVMs didn't make much of a difference (AP either remained unchanged or changed little). On the whole, it appears that my technique yields an overall improvement by helping only a relatively small fraction of topics, which is why we observe statistical significance even though the magnitude of the improvements is small. Nevertheless, the technique does appear to be reshuffling the ranked lists quite dramatically: Figure 1 compares rank correlation (measured using Kendall's τ) between the base and final runs at ranks {10, 20, 50, 100, 200, 500, 1000}. The box-and-whiskers plots show the per-topic distribution of the Kendall's τ values. As expected, the early ranks change little since those documents are used as pseudo-positive labels. However, beyond the early ranks, the "before" and "after" results are quite different. Interestingly, in most cases, this large reshuffling does not appear to change AP much.

6 CONCLUSIONS

One common thread in recent commentary referenced in the introduce is that complexity (in terms of models, parameter estimation, etc.) potentially obscures understanding, especially in the absence of rigorous ablation studies. Simplicity is a virtue, and simple yet effective techniques should be the most preferred type of solution overall. I believe my technique fits this description.

ACKNOWLEDGMENTS

This research was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

REFERENCES

- N. Abdul-Jaleel, J. Allan, W. Croft, F. Diaz, L. Larkey, X. Li, D. Metzler, M. Smucker, T. Strohman, H. Turtle, and C. Wade. 2004. UMass at TREC 2004: Novelty and HARD. In TREC.
- [2] T. Armstrong, A. Moffat, W. Webber, and J. Zobel. 2009. Improvements That Don't Add Up: Ad-hoc Retrieval Results Since 1998. In CIKM. 601–610.
- [3] C. Buckley. 2005. The SMART Project at TREC. In TREC: Experiments and Evaluation in Information Retrieval. MIT Press.

- [4] G. Cao, J.-Y. Nie, J. Gao, and S. Robertson. 2008. Selecting Good Expansion Terms for Pseudo-relevance Feedback. In SIGIR.
- [5] G. Cormack and M. Mojdeh. 2009. Machine Learning for Information Retrieval: TREC 2009 Web, Relevance Feedback and Legal Tracks. In TREC.
- [6] G. Cormack, M. Smucker, and C. Clarke. 2011. Efficient and Effective Spam Filtering and Re-ranking for Large Web Datasets. *Information Retrieval* 14, 5 (2011), 441–465.
- [7] W. Croft and D. Harper. 1979. Probabilistic Models of Document Retrieval with Relevance Information. *Journal of Documentation* 35, 4 (1979), 285–295.
- [8] F. Diaz. 2005. Regularizing Ad Hoc Retrieval Scores. In CIKM. 672-679.
- [9] M. Grossman and G. Cormack. 2017. MRG_UWaterloo and WaterlooCormack Participation in the TREC 2017 Common Core Track. In TREC.
- [10] N. Jardine and C. van Rijsbergen. 1971. The Use of Hierarchic Clustering in Information Retrieval. Information Storage and Retrieval 7, 5 (1971), 217–240.
- [11] S. Kharazmi, F. Scholer, D. Vallet, and M. Sanderson. 2016. Examining Additivity and Weak Baselines. ACM Transactions on Information Systems 34, 4 (2016), Article 23
- [12] C. Li, Y. Sun, B. He, L. Wang, K. Hui, A. Yates, L.Sun, and J. Xu. 2018. NPRF: A Neural Pseudo Relevance Feedback Framework for Ad-hoc Information Retrieval. In EMNLP, 4482–4491.
- [13] J. Lin. 2018. The Neural Hype and Comparisons Against Weak Baselines. SIGIR Forum 52, 2 (2018), 40–51.
- [14] Z. Lipton and J. Steinhardt. 2018. Troubling Trends in Machine Learning Scholarship. arXiv:1807.03341v2 (2018).
- [15] T.-Y. Liu. 2009. Learning to Rank for Information Retrieval. Foundations and Trends in Information Retrieval 3, 3 (2009), 225–331.
- [16] R. Nallapati. 2004. Discriminative Models for Information Retrieval. In SIGIR. 64–71
- [17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12, Oct (2011), 2825–2830.
- [18] K. Raman, R. Udupa, P. Bhattacharya, and A. Bhole. 2010. On Improving Pseudo-Relevance Feedback Using Pseudo-Irrelevant Documents. In ECIR. 573-576.
- [19] S. Robertson and K. Spark Jones. 1976. Relevance Weighting of Search Terms. JASIS 27, 3 (1976), 129–146.
- [20] J. Rocchio. 1971. Relevance Feedback in Information Retrieval. In The SMART Retrieval System—Experiments in Automatic Document Processing. Prentice-Hall.
- [21] I. Ruthven and M. Lalmas. 2003. A Survey on the Use of Relevance Feedback for Information Access Systems. Knowledge Engineering Review 18, 2 (2003), 95–145.
- [22] D. Sculley, J. Snoek, A. Wiltschko, and A. Rahimi. 2018. Winner's Curse? On Pace, Progress, and Empirical Rigor. In ICLR Workshops.
- [23] Z. Xu and R. Akella. 2008. A Bayesian Logistic Regression Model for Active Relevance Feedback. In SIGIR. 227–234.
- [24] R. Yan, A. Hauptmann, and R. Jin. 2003. Multimedia Search with Pseudo-relevance Feedback. In CIVR. 238–247.
- [25] P. Yang, H. Fang, and J. Lin. 2018. Anserini: Reproducible Ranking Baselines Using Lucene. Journal of Data and Information Quality 10, 4 (2018), Article 16.
- [26] P. Yang and J. Lin. 2019. Reproducing and Generalizing Semantic Term Matching in Axiomatic Information Retrieval. In ECIR.
- [27] R. Yu, Y. Xie, and J. Lin. 2019. Simple Techniques for Cross-Collection Relevance Feedback. In ECIR.

4