# PGT: Pseudo Relevance Feedback Using a Graph-Based Transformer

HongChien Yu(✉), Zhuyun Dai, and Jamie Callan

Carnegie Mellon University, Pittsburgh, USA
{hongqiay,zhuyund,callan}@cs.cmu.edu

**Abstract.** Most research on pseudo relevance feedback (PRF) has been done in vector space and probabilistic retrieval models. This paper shows that Transformer-based rerankers can also benefit from the extra context that PRF provides. It presents PGT, a graph-based Transformer that sparsifies attention between graph nodes to enable PRF while avoiding the high computational complexity of most Transformer architectures. Experiments show that PGT improves upon non-PRF Transformer reranker, and it is at least as accurate as Transformer PRF models that use full attention, but with lower computational costs.

## 1 Introduction

Pseudo relevance feedback (PRF) uses context defined by the top-ranked documents of an initial retrieval to improve a subsequent retrieval. Most prior research has been done in vector space [20], probabilistic [19], and language modeling [13,16,23] retrieval models.

Recently the field has moved to Transformer-based rerankers [18] that are more accurate and computationally complex. Most Transformer-based rerankers learn contextualized representations from query-document pairs, but they have two limitations. First, the query-document pair provides limited context for query understanding. Second, most Transformers have computational complexity quadratic to the input sequence length, rendering longer context infeasible.

To overcome these limitations, we propose a PRF method using a graph-based Transformer (PGT). PGT constructs a graph of the query, the candidate document, and the feedback documents. It uses intra-node attention to contextualize the query according to each individual document, and it uses inter-node attention to aggregate information. With the graph approach, PGT can utilize richer relevance context using a configurable number of feedback documents. Its inter-node attention is sparsified, so it also saves computation.

This paper makes two contributions to the study of pseudo relevance feedback in Transformer architectures. First, it investigates several ways of using PRF documents as context for Transformer rerankers. It shows that PGT improves upon non-PRF Transformer rerankers, and that PGT is at least as accurate as Transformer PRF models that use full attention, while reducing computation.

D. Hiemstra et al. (Eds.): ECIR 2021, LNCS 12657, pp. 440–447, 2021.
https://doi.org/10.1007/978-3-030-72240-1_46

Second, it studies the impact of contextual interactions by adjusting the configuration of the graph. It shows that token-level interaction between the query and feedback documents is critical, while document-level interaction is sufficient to aggregate information from multiple documents.

## 2 Related Work

Pseudo-relevance feedback is a well-studied method of generating more effective queries. Typically pseudo-relevance feedback uses the top-ranked documents to add query terms and set query term weights. Well-known methods include Rocchio [20], BM25 expansion [19], relevance models [13], and KL expansion models [16,23]. A large body of work studies which documents to use for expansion (e.g., [3]). Most methods were designed for discrete bag-of-words representations.

Recent research also studies PRF in neural networks. Li et al. [15] present a neural PRF framework that uses a feed forward network to combine the relevance scores of feedback documents. Only marginal improvement was observed over simple score summation, indicating that the framework does not make the best use of the feedback documents' information.

Recently, pre-trained Transformer [21] language models, such as BERT [6], have improved the state-of-the-art for ad hoc retrieval. Most Transformer-based rerankers are applied to individual query-document pairs. Some research explores jointly modeling multiple top retrieved documents in a Transformer architecture for question clarification [11], question answering [10,14] or code generation [8]. The effectiveness of using top retrieved documents in Transformer rerankers remains to be studied.

While the Transformer-based architectures have achieved state-of-the-art results in multiple natural language tasks [6], the original self-attention mechanism incurs computational complexity quadratic to the length of the input sequence. Therefore, much recent work studies sparsifying Transformer attention [1,2,24]. Among these models, Transformer-XH [24] features an underlying graph structure, where each node represents a text sequence, which makes it a good candidate for multi-sequence tasks such as PRF.

Transformer-XH employs full-attention within each sequence, but it sparsifies inter-sequence attention. Specifically, for each document sequence $s$, the $l$th layer encoder calculates the intra-sequence, token-level attention by the standard self-attention. Inter-sequence, document-level attentions are calculated using the hidden representations of each sequence's first token `[CLS]`:

$$\hat{h}^{l}_{s,0} = \sum_{s' \in \mathcal{N}(s)} softmax_{s'}\left(\frac{\hat{q}^{T}_{s,0} \cdot \hat{k}_{s',0}}{\sqrt{d_k}}\right) \cdot \hat{v}_{s',0}, \tag{1}$$

where $\mathcal{N}(s)$ are the neighboring document sequences of $s$ in the graph. This allows the `[CLS]` token to carry context from other neighboring sequences. Such information is propagated to other tokens in the sequence through the intra-sequence attention in the next layer. Hence Transformer-XH outputs a condensed representation that contains both the global graph-level information and the local sequence-level information.

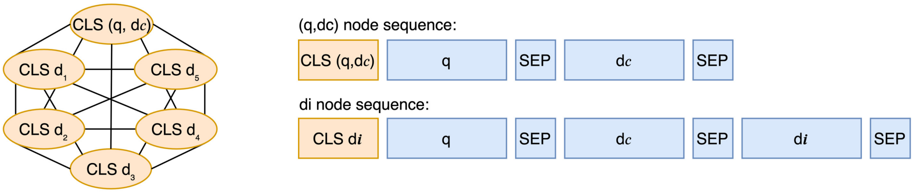

**Fig. 1.** Right: Nodes in PGT contextualize the query using the candidate document $d_c$ and the feedback documents $d_i$ with intra-sequence, token-level attention. Left: The input graph is fully connected with inter-sequence attention among [CLS] tokens.

## 3 Proposed Method

We propose PGT, a PRF reranker with a graph-based Transformer. Given a query $q$, a candidate document $d_c$, and feedback documents $d_1$, ..., $d_k$ retrieved by a first-stage retrieval algorithm, the goal is to predict the score of $d_c$ by aggregating information from feedback documents. To achieve this goal, PGT adopts the Transformer-XH [24] architecture, and builds a graph of $q$, $d_c$ and $d_1$, ..., $d_k$. Figure 1 illustrates the graph.

PGT has two types of nodes. The $d_i$ nodes contextualize the query using feedback documents. As shown in Fig. 1 (right), the input to a $d_i$ node is the text of $d_i$, with $q$ and $d_c$ prepended in order to extract information specific for predicting the relevance between $q$ and $d_c$. The input text sequence is fed into a Transformer module with standard token-level self-attention. To distinguish different parts of the input, we associate segment id 0 with $q$ and $d_c$, and 1 with $d_i$. In addition to the feedback document nodes, PGT also adopts a special node for the query-candidate pair $(q, d_c)$. The input of the $(q, d_c)$ node is the concatenation of the query and candidate document, which constitutes a typical input sequence to existing Transformer-based rerankers. We hypothesize that the $(q, d_c)$ node will help the model focus more on the query-candidate pair.

PGT aggregates sequence-level information through inter-sequence attention. Within the sequence, the Transformer encodes the [CLS] token to represent the whole sequence (Fig. 1 right). Between the sequences, all [CLS] tokens attend to each other to gather information from other sequences (Fig. 1 left). We follow Zhao et al. [24] and incorporate inter-sequence attention in the last three Transformer encoder layers. The model is trained on a binary relevance classification task using cross-entropy loss, and it predicts the final relevance score using a weighted sum of all the [CLS] representations [24].

## 4 Experimental Setup

This section describes our datasets, baselines and other experimental settings.

### 4.1 Datasets

Experiments were done with the MS MARCO Passage Ranking task dataset [17]. It contains about 8.8 million passages and about 0.5 million queries with relevance judgments as training data. Each query has an equal number of relevant and non-relevant passages. We used the official evaluation query set from the TREC 2019 Deep Learning Track [4]. It contains 43 test queries manually annotated by NIST on a four-point scale. On average, a query has 95 relevant documents. We report NDCG@10, MAP@10, and MAP@100.

### 4.2 Baselines

We compare PGT to initial rankers, a non-PRF reranker, and PRF models.

- **BM25 (initial ranker):** We used Anserini's implementation [22]. $k_1$ and $b$ were tuned using a parameter sweep on 500 training queries, following [5].
- **CLEAR (initial ranker):** This model combines BM25's lexical retrieval and BERT's dense embedding retrieval. It performs significantly better than BM25 on our dataset. We used the rankings provided by Gao, et al. [7].
- **BERT reranker (non-PRF reranker):** This is a standard BERT reranker, whose input is the concatenated sequence of the query $q$ and the candidate document $d_c$. We trained the model following Nogueira and Cho [18].
- **RM3 (PRF):** This is a traditional language modeling PRF method [12,13].
- **BERT PRF (PRF):** This is the same as BERT reranker except that we concatenate $(q, d_c, d_1, d_2, ..., d_k)$ to form a PRF input sequence, with documents separated by `[SEP]`. Limited by the input length constraint of BERT [6], we used 5 feedback documents. Same as for PGT, we used segment id 0 for $q$ and $d_c$, and 1 for $d_i$.

### 4.3 PGT Graph Variants

Modeling queries and documents in a graph gives control over how representations are contextualized. We examined 5 graph variants to study this effect.

- **PGT base** is the graph described in Sect. 3. The query is first contextualized by the candidate and feedback document at the token-level. Feedback information is then aggregated following the graph structure. The $(q, d_c)$ node emphasizes $q$ and $d_c$ at the graph-level. This variant has the richest context.
- **PGT w/o pre $\boldsymbol{d_c}$** removes prepended candidate from the $d_i$ nodes, so each query is only contextualized by the feedback document at sequence-level.
- **PGT w/o pre $\boldsymbol{q, d_c}$** removes both the prepended query and the prepended candidate from the feedback nodes. Each feedback document hence only contextualizes the query at the graph-level.

- **PGT w/o node $\boldsymbol{d_c}$** removes candidate from the $(q, d_c)$ node, so only $q$ is emphasized again at the graph-level.
- **PGT w/o node $\boldsymbol{q, d_c}$** removes the $(q, d_c)$ node from the graph, so $q$ and $d_c$ are not emphasized again at the graph-level.

### 4.4 Training and Evaluation

We implement PGT based on the Transformer-XH [24] PyTorch implementation. The parameters for the intra-sequence attention are initialized from a pre-trained BERT base model [6], and those for the inter-sequence attention are initialized according to Xavier et al. [9]. We train the model for 2 epochs, with per-GPU batch size = 4 on 2 GPUs. The maximum node sequence length is 128, and the learning rate is 5e-6 with linear decay.

We train both BERT PRF and PGT using feedback documents from BM25. In order to test how Transformer-based PRF models generalize when different initial rankers are used, we evaluate them using both BM25 and CLEAR. We follow prior research [7,18] and report the results at each model's best reranking depth $r$ (Table 1).

## 5 Experimental Results

**PRF vs. non-PRF Transformers.** We study the effectiveness of PRF in Transformer-based models by comparing PGT and BERT PRF with BERT reranker. Table 1 shows that all PRF Transformers outperform BERT reranker on MAP@10 using either initial ranker. In particular, PGT achieves MAP@10 13.0% and 7.4% better than BERT reranker on BM25 and CLEAR respectively, with comparable NDCG@10. The results suggest that the richer context provided by PRF helps Transformers rank relevant documents to the very top.

PRF enables Transformers to exploit high-quality initial rankings better. Comparing BM25 and CLEAR results in Table 1, we found that when the initial ranker is stronger, PGT achieves the best performance across all metrics, closely followed by BERT PRF. In comparison, BERT reranker cannot make the best of the initial retrieval of CLEAR, as reported by prior research [7].

**PGT vs. BERT PRF.** While PGT rankings are at least as good as BERT PRF, it is more computationally efficient. Using $k = 5$ for a fair comparison, we calculated the number of multiplication and addition operations. PGT consumes 88% as many operations on each input example compared with BERT PRF. In addition, PGT requires smaller reranking depth (Table 1). Using BM25 as the initial ranker, the computational cost is hence only 44% of BERT PRF's.

Compared with BERT PRF, PGT allows flexible configurations on the graph structure (Table 1). As discussed in Sect. 4.3, the graph structure controls how relevance context flows across the graph. Contrary to our initial intuition, removing the $(q, d_c)$ node partially or entirely (PGT w/o node $d_c$ and PGT w/o node $q, d_c$) achieves the best results among all graph variants. $q$ is an impoverished description of the information need compared to feedback documents $d_1 \ldots d_k$, which may explain why the comparison of $q$ to $d_c$ is less useful than comparisons between $d_c$ and high-quality documents.

The number of feedback documents $k$ is a parameter that is usually tuned. BERT's self-attention mechanism restricts the input sequence length, limiting BERT to 5 feedback documents on our dataset. PGT has no such restriction. Our experiments use $k = 7$ for PGT because it is more effective (Table 2).

**Table 1.** The evaluation results with BM25 and CLEAR as initial rankers. RM3 is shown for completeness, but it is not competitive, so it is not discussed. We report the results at each models' best reranking depth ($r$) according to prior research [7,18]. We use $k = 7$ feedback documents for PGT. $*$ and $\dagger$ indicate statistical significance over the initial ranker and BERT reranker using t-test with $p \leq 0.05$.

| | BM25 | | | | CLEAR | | | |
|---|---|---|---|---|---|---|---|---|
| | NDCG | MAP | MAP | | NDCG | MAP | MAP | |
| | @10 | @10 | @100 | r | @10 | @10 | @100 | r |
| Initial ranker | 0.5058 | 0.1126 | 0.2993 | – | 0.6990 | 0.1598 | 0.4181 | – |
| RM3 | 0.5180 | 0.1192 | 0.3370$^*$ | 1K | –[a] | – | – | – |
| BERT Reranker | 0.6988$^*$ | 0.1457$^*$ | 0.3905$^*$ | 1K | 0.7127 | 0.1572 | 0.4134 | 20 |
| BERT PRF | 0.6862$^*$ | 0.1495$^*$ | **0.4075**$^*$ | 1K | 0.7188 | 0.1646 | 0.4203 | 20 |
| PGT base | 0.6712$^*$ | 0.1542$^*$ | 0.3927$^*$ | 500 | 0.7238$^*$ | 0.1660 | 0.4205 | 20 |
| PGT w/o pre $d_c$ | 0.6693$^*$ | 0.1523$^*$ | 0.3563$^*$ | 500 | 0.7146 | 0.1658 | 0.4194 | 20 |
| PGT w/o pre $q, d_c$ | 0.6676$^*$ | 0.1468$^*$ | 0.3450$^*$ | 500 | 0.7005 | 0.1572 | 0.4145 | 20 |
| PGT w/o node $d_c$ | 0.6840$^*$ | 0.1586$^*$ | 0.3868$^*$ | 500 | 0.7139 | **0.1689**$^*$ | 0.4192 | 20 |
| PGT w/o node $q, d_c$ | **0.7078**$^*$ | **0.1646**$^{*\dagger}$ | 0.3819$^*$ | 500 | **0.7326**$^*$ | 0.1654 | **0.4220** | 20 |

[a] CLEAR jointly trains a hybrid of sparse and dense retrieval models. Running RM3 on CLEAR is an open question that is beyond the scope of this work.

**Table 2.** PGT base using different numbers of feedback documents ($k$)

| | BM25 | | | CLEAR | | |
|---|---|---|---|---|---|---|
| | NDCG | MAP | MAP | NDCG | MAP | MAP |
| k | @10 | @10 | @100 | @10 | @10 | @100 |
| 5 | 0.6344 | 0.1497 | 0.3536 | 0.6923 | 0.1653 | 0.4177 |
| 7 | **0.6712** | **0.1542** | 0.3927 | **0.7238** | **0.1660** | **0.4205** |
| 9 | 0.6538 | 0.1476 | **0.3931** | 0.6940 | 0.1636 | 0.4180 |

## 6 Conclusion

Most Transformer-based rerankers learn contextualized representations for query-document pairs, however queries are impoverished descriptions of information needs. This paper presents PGT, a pseudo relevance feedback method that uses a graph-based Transformer. PGT graphs treat feedback documents as additional context and leverage sparse attention to reduce computation, enabling them to use more feedback documents than is practical with BERT-based rerankers.

Experiments show that PGT improves upon non-PRF BERT rerankers. Experiments also show that PGT rankings are at least as good as BERT PRF rerankings, however they are produced more efficiently due to fewer computations per document and fewer documents reranked per query. PGT is robust, delivering effective rankings under varied graph structures and with two rather different initial rankers.

## References

1. Beltagy, I., Peters, M.E., Cohan, A.: Longformer: the long-document transformer. arXiv preprint arXiv:2004.05150 (2020)
2. Child, R., Gray, S., Radford, A., Sutskever, I.: Generating long sequences with sparse transformers. arXiv preprint arXiv:1904.10509 (2019)
3. Collins-Thompson, K., Callan, J.: Estimation and use of uncertainty in pseudo-relevance feedback. In: Proceedings of the Thirtieth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (2007)
4. Craswell, N., Mitra, B., Yilmaz, E., Campos, D., Voorhees, E.M.: Overview of the TREC 2019 deep learning track. arXiv preprint arXiv:2003.07820 (2020)
5. Dai, Z., Callan, J.: Context-aware term weighting for first stage passage retrieval. In: Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, pp. 1533–1536. ACM (2020)
6. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 4171–4186. ACL (2019)
7. Gao, L., Dai, Z., Chen, T., Fan, Z., Durme, B.V., Callan, J.: Complement lexical retrieval model with semantic residual embeddings. arXiv preprint arXiv:2004.13969 (2020)
8. Gemmell, C., Rossetto, F., Dalton, J.: Relevance transformer: generating concise code snippets with relevance feedback. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 2005–2008. ACM (2020)
9. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010. JMLR Proceedings, vol. 9, pp. 249–256. JMLR.org (2010)
10. Guu, K., Lee, K., Tung, Z., Pasupat, P., Chang, M.: REALM: retrieval-augmented language model pre-training. arXiv preprint arXiv:2002.08909 (2020)
11. Hashemi, H., Zamani, H., Croft, W.B.: Guided transformer: leveraging multiple external sources for representation learning in conversational search. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1131–1140 (2020)
12. Jaleel, N.A., et al.: Umass at TREC 2004: novelty and HARD. In: Proceedings of the Thirteenth Text REtrieval Conference, TREC 2004. NIST Special Publication, vol. 500–261. NIST (2004)

13. Lavrenko, V., Croft, W.B.: Relevance-based language models. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 120–127. ACM (2001)
14. Lee, K., Chang, M., Toutanova, K.: Latent retrieval for weakly supervised open domain question answering. In: Proceedings of the 57th Conference of the Association for Computational Linguistics, vol. 1, pp. 6086–6096. ACL (2019)
15. Li, C., et al.: NPRF: a neural pseudo relevance feedback framework for ad-hoc information retrieval. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 4482–4491. ACL (2018)
16. Lv, Y., Zhai, C.: Revisiting the divergence minimization feedback model. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, pp. 1863–1866. ACM (2014)
17. Nguyen, T., et al.: MS MARCO: a human generated machine reading comprehension dataset. In: Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016). CEUR Workshop Proceedings, vol. 1773. CEUR-WS.org (2016)
18. Nogueira, R., Cho, K.: Passage re-ranking with BERT. arXiv preprint arXiv:1901.04085 (2019)
19. Robertson, S.E., Zaragoza, H.: The probabilistic relevance framework: BM25 and beyond. Found. Trends Inf. Retr. **3**(4), 333–389 (2009)
20. Rocchio, J.J.: Relevance feedback in information retrieval. In: The SMART Retrieval System - Experiments in Automatic Document Processing, pp. 313–323. Prentice-Hall, Englewood Cliffs (1971)
21. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems, vol. 2017. pp. 5998–6008 (2017)
22. Yang, P., Fang, H., Lin, J.: Anserini: enabling the use of lucene for information retrieval research. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1253–1256. ACM (2017)
23. Zhai, C., Lafferty, J.D.: Model-based feedback in the language modeling approach to information retrieval. In: Proceedings of the 2001 ACM CIKM International Conference on Information and Knowledge Management, pp. 403–410. ACM (2001)
24. Zhao, C., Xiong, C., Rosset, C., Song, X., Bennett, P.N., Tiwary, S.: Transformer-XH: multi-evidence reasoning with extra hop attention. In: 8th International Conference on Learning Representations, ICLR 2020. OpenReview.net (2020)