Vector Space Interpolation for Query Expansion

Deepanway Ghosal[1], **Somak Aditya**[2], **Sandipan Dandapat**[3], **Monojit Choudhury**[4]

[1] ISTD, Singapore University of Technology and Design, [2] Department of CSE, IIT Kharagpur
[3] Microsoft R&D, India, [4] Turing India, Microsoft
deepanway_ghosal@mymail.sutd.edu.sg
saditya@cse.iitkgp.ac.in
{sadandap, monojitc}@microsoft.com

Abstract

Topic-sensitive query set expansion is an important area of research that aims to improve search results for information retrieval. It is particularly crucial for queries related to sensitive and emerging topics. In this work, we describe a method for query set expansion about emerging topics using vector space interpolation. We use a transformer model called OPTIMUS, which is suitable for vector space manipulation due to its variational autoencoder nature. One of our proposed methods – *Dirichlet interpolation* shows promising results for query expansion. Our methods effectively generate new queries about the sensitive topic by incorporating set-level diversity, which is not captured by traditional sentence-level augmentation methods such as paraphrasing or back-translation.

1 Introduction

In web-search scenario, users may input queries that are not offensive (or controversial) by themselves; but may *leak* controversial queries through auto-suggest or return offensive (or controversial) documents in the search engine results pages (SERP). These queries, denoted as *threat* queries, often pertain to sensitive topics, such as anti-semitism, and climate change. To minimize such inadvertent *leakage*, search engine companies identifies a list of potentially sensitive topics that such queries pertain to. Then they build topic-wise classifiers to categorize queries into such topics, so that queries from sensitive *topics* can be handled specifically (such as post-processing the SERP results). Formally, a *topic* is defined as an emerging subarea of a broader sensitive issue that leads to adverse public relations issues and user experience. Often emerging topics and trends become a hive of threat queries, for instance, the 2020 USA elections, and COVID19 vaccination. Such topics could be time-sensitive as well (such as Michael Jackson's death, black lives matter movement). Annotating a sizable amount of training data for these topics is challenging due to time-sensitivity, and ever-growing number of the topics. Hence, our central motivation is to propose an efficient topic-sensitive query set expansion technique from a small set of annotated queries for emerging topics and trends.

Given a small number of queries about a sensitive topic, we aim to automatically generate a larger extensive set of queries while maintaining topical consistency. This expanded set could be later used to improve the classifier / post-processing techniques to provide better search results. Initially, human annotators tag this initial set of queries (denoted as *seed* queries) as belonging to the a sensitive topic. The seed query set is assumed to be small (typically around ten queries), as it is impractical for annotators to create a large set of seed queries with the increasing number of sensitive topics. In literature, query expansion is performed using paraphrasing (Zukerman and Raskutti, 2002; Figueroa and Neumann, 2013), and template transformation (Gu et al., 2019). However, these methods attempt to preserve the semantic meaning of the original query, which is not our intent. Although some work have focused on improving the sentence-level diversity of the paraphrases (Park et al., 2019; Xu et al., 2018), they are unable to broaden the coverage of the seed queries, where set-level diversity and generalization are crucial. In this paper, we propose a query expansion method based on vector space interpolation. We consider the interaction between the queries in the seed set that leads to extensive and completely new queries (within the sensitive topic), which is impossible to generate using traditional text augmentation techniques, such as paraphrasing or back-translation.

Our contributions are as follows: i) we propose a method for set expansion using vector space interpolation (specifically Dirichlet interpolation) which ii) ensures set-level diversity and generalizes to create new queries within the given topic.

Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 405–410
November 20–23, 2022.

2 Vector Space Interpolation

We use the variational autoencoder language model OPTIMUS (Li et al., 2020) for vector space interpolation. It consists of a BERT encoder (Devlin et al., 2019) and a GPT2 decoder (Radford et al.). The encoder and the decoder are linked through a latent connector, which serves as the bottleneck layer of the autoencoder. It is pre-trained on a large text corpus, where it learns to reconstruct a sentence x from the decoder, given that sentence x as input to the encoder.

OPTIMUS learns to organize sentences according to high-level semantics (topic, sentiment, tense, etc.) in the intermediate latent space through the pre-training objective. Thus, it allows easy manipulation of the dense sentence representations in the latent vector space. As argued in Li et al. (2020), this is possible because: i) sentences are densely represented in the latent space as a result of pre-training, and ii) KL regularization of the VAE and the continuity property of neural networks allow latent vectors with similar semantics to be smoothly organized together. It is thus possible to combine two or more sentences by performing vector operations over their latent representations. The resultant vector could then be used to perform controlled generation through the decoder.

We denote a topic t as consisting of seed queries $s_1, s_2, .., s_n$. The latent vectors from OPTIMUS corresponding to the seed queries are $z_1, z_2, .., z_n$. We combine the latent vectors in different ways to create the modified latent vector z. The modified latent vector z is then used in the decoder to generate new synthetic queries. We use the following methods to manipulate the latent vectors of the seed queries:

Linear Interpolation: The interpolation technique used in Li et al. (2020). The modified latent vector z is created from a linear combination of two vectors, where the weights sum to 1.

$$z = q * z_i + (1 - q) * z_j$$

where, $q \in [0, 1]$, $i, j \in [1, 2, .., n]$, and $i \neq j$. In particular, we use $q = [0.1, 0.2, .., 0.9]$ in increments of 0.1. We consider all $\binom{n}{2}$ possible combinations of i and j from the n seed queries.

Polar Interpolation: If seed queries s_i and s_j are not very similar, then their latent vectors z_i and z_j are observed to be roughly orthogonal having roughly the same Euclidean norms. Thus, the modified latent vector z obtained from the linear interpolation between z_i and z_j has a different norm than either z_i or z_j. As q changes in increments of 0.1 between 0 and 1, the norm of z becomes proportional to $\sqrt{q^2 + (1-q)^2}$. The topicality, quality of generated queries thus become poorer when z is sent to the decoder, as the OPTIMUS decoder expects a vector whose norm is similar to z_i or z_j.

The decoder performs better when the interpolated vector has the same norm as z_i or z_j. We use Cartesian to polar co-ordinate transfer of the weights to achieve this property. We term this method as polar interpolation. The latent vector z is created from a linear combination of z_i and z_j, where the square of weights sum to 1.

$$z = \sqrt{q} * z_i + \sqrt{1 - q} * z_j$$

where, $q \in [0, 1]$, $i, j \in [1, 2, .., n]$, $i \neq j$. We use the same choices of q, i, j as in linear interpolation.

Dirichlet Interpolation: The Dirichlet interpolation method is a more expressive interpolation technique that uses all seed queries from topic t to create the latent vector z. Compared to linear and polar interpolation (which uses two queries at a time), the Dirichlet interpolation creates more diverse latent vectors z, resulting in a more expressive expanded query set. We create the latent vector z for Dirichlet interpolation as follows:

$$z = \sum_{k=1}^{n} \sqrt{q_k} * z_k = \sqrt{q_1} * z_1 + \cdots + \sqrt{q_n} * z_n$$

where, $q_1 + q_2 + \cdots + q_n = 1$, and $q_k > 0 \; \forall \; k$. The probability density function of the Dirichlet distributed random vector $\mathcal{Q}$ satisfies the following: $p(q) \propto \prod_{i=1}^{n} q_i^{\alpha_i - 1}$, where α is a n-dim vector containing the positive concentration parameters.

3 Methodology

3.1 Query OPTIMUS

The original OPTIMUS models was trained on sentences from the English Wikipedia and optionally the SNLI dataset (Bowman et al., 2015). We found that interpolation is more effective for queries when the OPTIMUS model is further pre-trained on a query-specific corpus. We start with the Wikipedia and SNLI pre-trained checkpoint of OPTIMUS having a latent size of 768 and β of 0.5. β specifies the KL regularization strength during training. We continue training from this checkpoint with the originally proposed objective functions of OPTIMUS

on the queries of the MS MARCO dataset (Nguyen et al., 2016) for 3 epochs with a β of 1. We denote this model as the Query OPTIMUS model.

3.2 Interpolation

Given a topic t with n seed queries $s_1, s_2, .., s_n$, we use all pair combinations of seed queries for linear and polar interpolation. We use $q = [0.1, 0.2, .., 0.9]$ in increments of 0.1 to obtain the values of weights for the linear and polar combination. In total, we create $9 \times \binom{n}{2}$ latent vectors and corresponding decoded outputs.

For Dirichlet interpolation, we select n integers randomly (with replacement) between 1 and $50 \times n$. This n integers constitutes the n-dim concentration vector α for the Dirichlet distribution. We repeat this process $9 \times \binom{n}{2}$ times to create the same number of decoded outputs from Dirichlet interpolation as the linear and polar interpolation. The upper range of $50 \times n$ is a choice of hyperparameter which worked well for our experiments.

3.3 Post-Processing

The Query OPTIMUS decoded outputs are not always grammatically correct or well-formed English sentences or queries. We use a grammatical error correction model and a paraphrasing model to rectify the outputs of the decoder. Both are T5-Large (Raffel et al., 2020) models trained on respective task-specific parallel corpora.

4 Experimental Study

4.1 Query Topics and Evaluation Strategy

We use 15 sensitive topics of queries about emerging issues such as USA elections, politicians, COVID-19, vaccination, social media bans, etc. We use a combination of automatic and human evaluation metrics to measure the quality of synthetically generated queries.

4.1.1 Automatic Evaluation

We design an evaluation setup to measure the topical consistency, diversity, and quality of generated queries. The following metrics are used:

Topical Consistency: The generated queries should belong to the topic of the seed queries. We measure topical consistency using dense sentence embeddings from the *all-mpnet-base-v2* model (Song et al., 2020; Reimers and Gurevych, 2019) as follows: (i) The average euclidean distance of the generated query embeddings from the seed query embeddings. A lower value indicates closer to the original topic implying more topical consistency. The metric is denoted as **D-Avg**; (ii) The average cosine similarity of the generated query embeddings with the seed query embeddings. A higher value indicates more topical consistency. The metric is denoted as **CS-Avg**.

Diversity: The synthetically generated queries should ideally form a diverse set. This is a desired quality as we do not want the generated set to have repetitions or have elements very close to each other. We measure the diversity of the generated set using the **Self-BLEU** (Zhu et al., 2018) metric. We measure the average BLEU between all pairs of the queries in the generated set, and denote it as the Self-BLEU score. We compute Self-BLEU over uni-gram and bi-grams.

Quality: The generated queries should ideally have qualitative properties of human written queries and more generally of natural language. In other words, the generated queries should be well-formed query-like, such that they could be useful in downstream applications. We use the following metrics for automatic query quality evaluation: (i) The query well-formedness score or **QWF** score aims to measure whether the generated query is well-formed. We use a RoBERTa-base model (Liu et al., 2019) trained on the query well-formedness dataset (Faruqui and Das, 2018) to measure the score; (ii) Pretrained language models trained using the masked language modelling (MLM) objective can also be used to score sentences or queries. We use the method proposed by Salazar et al. (2020) to score a sentence with the RoBERTa-base model using pseudo-log-likelihood scores. We denote the metric as **MLM** score. A lower score is better; (iii) We also use the **GRUEN** score (Zhu and Bhat, 2020) for measuring linguistic quality of the generated queries. The metric is computed by considering grammaticality, non redundancy, focus, structure and coherence of the generated text.

4.1.2 Human Evaluation

We consider the generations from polar and Dirichlet interpolation method for human evaluation. We sample 250 queries from each of the 15 topics for human evaluation. To ensure diversity of the sampled queries we use the following method for each topic: i) 125 instances sampled based on sentences embeddings of the generated queries. We cluster all the generated queries of into 5 groups and then randomly sample from each group proportional to the group size. ii) We perform hierarchical clus-

Method	Automatic Evaluation							Human Evaluation		
	D-Avg ↓	CS-Avg ↑	S-BLEU1 ↓	S-BLEU2 ↓	QWF ↑	MLM ↓	GRUEN ↑	Overall ↑	Topic ↑	Grammar ↑
Linear	1.132	0.344	**0.127**	**0.027**	0.782	1.782	0.724	-	-	-
Polar	1.087	0.392	0.146	0.039	0.794	1.711	0.740	2.284	0.871	0.542
Dirichlet	**1.026**	**0.461**	0.294	0.141	**0.798**	**1.363**	**0.763**	**2.667**	**1.058**	**0.551**
PP+BT	0.899	0.568	0.214	0.087	0.524	2.415	0.650	-	-	-

Table 1: Results of automatic and human evaluation. ↑ and ↓ indicates higher and lower scores are better, respectively, among the three interpolation methods. S-BLEU indicates Self-BLEU scores. PP+BT represents the paraphrasing and back-translation baseline method. We merged paraphrased and back translated queries in a single set and performed evaluation.

tering based on BLEU distance between all pairs of generated queries apart from the ones sampled in the previous step. We then sample from each cluster proportional to its size such that the total number of sampled instances is 125.

We ask the human annotators to rate each of the 250 sampled queries of a topic on a scale of 0-5 based on topical consistency and well-formedness. The scale is as follows: does not belong to topic and not well-formed (0) or well-formed (1); belongs to a broader topic and not well-formed (2) or well-formed (3); belongs to the exact topic and not well-formed (4) or well-formed (5).

4.2 Results

We report results for automatic evaluation in Table 1. The Dirichlet interpolation method creates the most topical and highest quality generations as observed in the D-Avg, CS-Avg, QWF, MLM, and GRUEN scores. However, Dirichlet interpolation generated queries are less diverse than linear and polar interpolation generated queries. We hypothesize this is because of the averaging effect of all the seed queries in Dirichlet interpolation. We also surmise that a different method of choosing the concentration vector α could provide more diverse generations while maintaining the topical consistency and quality. Generated queries have QWF score of at-least 0.78 and GRUEN score of at-least 0.72, indicating satisfactory well-formedness and linguistic quality for all the interpolation methods.

The linear interpolation method provides the highest diversity among the generated queries, as indicated by the lowest Self-BLEU scores. However, it comes at the cost of topical inconsistency, where many generations are observed to become out of topic. Thus, the scores corresponding to the diversity metric in linear interpolation do not provide a complete interpretation of the results. Considering all the metrics, we conclude that the Dirichlet interpolation method performs the best, followed by polar and linear interpolation.

Topic: Mail in ballots election night. **Seed Queries:** 1) Fraud in counting mail in ballots; 2) Mail in ballots election night; 3) Mail in ballots used to steal election; 4) When are mail in ballot counted; 5) Election week because of mail in ballots; 6) Covid delaying mail in ballot counting; 7) Mail in ballot processing time
Dirichlet Interpolation Generated Queries: 1) Election integrity commission because of irregularities in results; 2) Ballots are counted after mail-in votes are cast; 3) COVID-19 mail-in ballot lookup; 4) Unintended problems because of mail-in ballots; 5) Fraud in counting mail-in postal codes; 6) Number of fraudulent voters; 7) Ban on mail-in voting and phony ballots; 8) Mail-in ballot processing can be tracked; 9) Mail-in ballot missing; 10) Fraud in counting the number of votes in USA; 11) Ballots with torn mail are counted; 12) COVID-19 delaying decision in NJ; 13) Illegal ballots sent to steal election; 14) COVID-19 illegal fraud in ballot counting
Paraphrased and Back-translated Queries: 1) Fraud in counting letters in ballot papers; 2) Post in ballot boxes Electoral night; 3) Post used in the ballot papers to steal election; 4) When will the post be counted in the election; 5) Election week due to postal ballot; 6) The mail was delayed in the ballot counting; 7) Ballot processing takes a long time

Table 2: Generated queries with Dirichlet interpolation, paraphrasing and back-translation from a given topic.

We merged paraphrased and back-translated queries in a single set and evaluated with our automatic evaluation metrics. The automatic evaluation results for this baseline method are shown in the PP+BT row in Table 1. The queries generated through this method are qualitatively (QWF, MLM, GRUEN) much poorer than all the interpolation methods. One interesting aspect is the topicality metric, where this method achieves the lowest (D-Avg) and highest (CS-Avg) scores. A better score is expected for this method as each generated query stays almost too close to one of the queries in the seed set. However, this is not useful in practice, as we want some amount of diversity and exploration in the expanded set. The interpolation techniques provide interesting compositions of concepts among the seed queries, resulting in much more diverse queries outside of the seed set, which is not possible with the paraphrasing and back-translation method. We show examples of Dirichlet, paraphrased and back-translated queries in Table 2. The majority of the generations are new set-level diverse queries strongly inclined to the

topic of the seed queries. Interpolation generated queries are also significantly more diverse, expressive, and extensive compared to paraphrased and back-translated queries. Given that the interpolation generated queries stay within the topic and the paraphrasing, back-translation baseline is unable to convey meaningful information beyond the seed set, we concluded that the interpolation technique is better and practically more useful.

We also report results for human evaluation in Table 1. We report the score (in 0-5 scale) averaged across the 15 topics as the *overall* score. We also report the disentangled *topic* score on a scale of 0-2 and *grammar* score on a scale of 0-1 in. The results suggest that the Dirichlet interpolation method is superior to the polar interpolation method across all the evaluation metrics. In particular, there is a significant improvement in topical consistency for Dirichlet interpolation, which leads to a 7% improvement in the overall score metric.

5 Conclusion

In this paper, we proposed a method for query expansion using different vector space interpolation techniques. We use the OPTIMUS variational autoencoder language model to perform the task of query expansion using linear, polar, and Dirichlet interpolation methods. We also propose several automatic and human evaluation metrics to compare the different interpolation techniques. The Dirichlet interpolation method shows the strongest results and is able to create set-level diverse queries about the given sensitive or emerging topic.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Manaal Faruqui and Dipanjan Das. 2018. Identifying well-formed natural language questions. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 798–803.

Alejandro Figueroa and Günter Neumann. 2013. Learning to rank effective paraphrases from query logs for community question answering. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*.

Yunfan Gu, Yang Yuqiao, and Zhongyu Wei. 2019. Extract, transform and filling: A pipeline model for question paraphrasing based on template. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 109–114, Hong Kong, China. Association for Computational Linguistics.

Chunyuan Li, Xiang Gao, Yuan Li, Xiujun Li, Baolin Peng, Yizhe Zhang, and Jianfeng Gao. 2020. Optimus: Organizing sentences via pre-trained modeling of a latent space. In *EMNLP*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. In *CoCo@ NIPs*.

Sunghyun Park, Seung-won Hwang, Fuxiang Chen, Jaegul Choo, Jung-Woo Ha, Sunghun Kim, and Jinyeong Yim. 2019. Paraphrase diversification using counterfactual debiasing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6883–6891.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Julian Salazar, Davis Liang, Toan Q Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems*, 33:16857–16867.

Jingjing Xu, Xuancheng Ren, Junyang Lin, and Xu Sun. 2018. Diversity-promoting gan: A cross-entropy based generative adversarial network for diversified text generation. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 3940–3949.

Wanzheng Zhu and Suma Bhat. 2020. Gruen for evaluating linguistic quality of generated text. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 94–108.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1097–1100.

Ingrid Zukerman and Bhavani Raskutti. 2002. Lexical query paraphrasing for document retrieval. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

A Geometric Interpretation of Interpolation

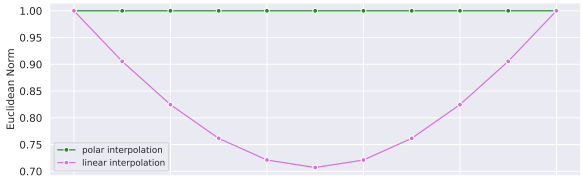

Figure 1

We illustrate the geometric interpretation of linear vs. polar (and by extension Dirichlet) interpolation in Fig. 1. We assume that the two corner points represent two unit vectors between which interpolation is performed. As evident in the figure, the euclidean norm of the linear interpolated vectors changes significantly in the intermediate steps. However, the norm of polar interpolated vectors maintain the unit norm. We empirically showed earlier that the polar (and Dirichlet) interpolation are better than linear interpolation for expanded query set generation. We attribute this to the difference in norm invariance property of polar and linear interpolation.

B Experimental Details

All the topics and queries used in this paper are in English language. The list of the 15 topics used in the experiments are as follows: i) Mail in ballots election night, ii) Election hacking, iii) Russian interference in elections, iv) Donald Trump and Taxes, v) Donald Trump social media bans, vi) Joe Biden forgets pledge, vii) US Citizenship of Kamala Harris, viii) Kamala Harris president eligibility, ix) COVID Threats: Florida deletes COVID data, x) Mask mandate repealed, xi) Fake COVID vaccination cards online, xii) Immune system is sufficient and vaccines not needed, xiii) Lockdowns not needed if vaccines actually work, xiv) Vaccination and infertility, and xv) Critical race theory.

We use a single Quadro RTX 8000 GPU for our experiments. It takes around 15 minutes to generate the expanded set using each interpolation technique for a topic with 10 seed queries.