Neural Disentanglement of Query Difficulty and Semantics

Sara Salamat
sara.salamat@torontomu.ca
Toronto Metropolitan University

Negar Arabzadeh
narabzad@uwaterloo.ca
University of Waterloo

Shirin Seyedsalehi
shirin.seyedsalehi@torontomu.ca
Toronto Metropolitan University

Amin Bigdeli
abigdeli@torontomu.ca
Toronto Metropolitan University

Morteza Zihayat
mzihayat@torontomu.ca
Toronto Metropolitan University

Ebrahim Bagheri
bagheri@torontomu.ca
Toronto Metropolitan University

ABSTRACT

Researchers have shown that the retrieval effectiveness of queries may depend on other factors in addition to the semantics of the query. In other words, several queries expressed with the same intent, and even using overlapping keywords, may exhibit completely different degrees of retrieval effectiveness. As such, the objective of our work in this paper is to propose a neural disentanglement method that is able to disentangle query semantics from query difficulty. The disentangled query semantics representation provides the means to determine semantic association between queries whereas the disentangled query difficulty representation would allow for the estimation of query effectiveness. We show through our experiments on the query performance prediction; and, query similarity calculation tasks that our proposed disentanglement method is able to show better performance compared to the state of the art.

CCS CONCEPTS

• **Information systems** → *Retrieval models and ranking*.

KEYWORDS

Information retrieval, Query performance prediction, Disentanglement

ACM Reference Format:
Sara Salamat, Negar Arabzadeh, Shirin Seyedsalehi, Amin Bigdeli, Morteza Zihayat, and Ebrahim Bagheri. 2023. Neural Disentanglement of Query Difficulty and Semantics. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3583780.3615189

1 INTRODUCTION

Deep neural networks have been increasingly used for various tasks in information retrieval and have shown impressive performance improvements over traditional retrieval methods [5, 26, 28, 34, 37]. This is primarily due to their ability to learn latent distributions of data through dense representations. For instance, in the context of ad hoc retrieval, dense neural rankers learn representations that effectively connect the query space to the document space and facilitate the retrieval of relevant documents for an input query [21, 35, 41]. Researchers have already shown that learnt representations encode a range of information without discriminating between them as long as the final representation is effective for the task at hand [12, 45]. In many cases, it is not immediately clear what each sub-part of the representation stands for and whether they carry any semantics independently. It is likely, as shown by earlier work, that representations of queries and documents may be amalgamating different attributes such as style, content semantics, tense, among others, without distinguishing between them [19, 20, 43].

Existing research has explored ways through which intertwined attributes in neural representations are separated. This process is referred to as disentanglement and has found application in areas such as controlled text generation [18], and style transfer [46], to name a few. The neural disentanglement process appears to be especially relevant to the task of ad hoc retrieval and how neural representations of queries are used [17, 25, 30]. There have been several studies showing similar queries that carry similar semantics or even at times are expressed using overlapping terminology, but with different ordering or tone, may end up producing completely different retrieval outcomes and hence have different retrieval effectiveness. For instance, the Matches Made in Heaven (MMH) dataset shows that there are over 180k queries in the 500k queries of the MS MARCO passage retrieval dataset where a small variation of the query can change retrieval effectiveness of wide range of queries from a mean average precision of 0.139 to 1 [1]. While query and document semantics are key in the retrieval process, the effectiveness of the query relies on additional factors that determine how difficult it is for the retrieval method to satisfy the query. This work is motivated by the evidence observed in at least 180k queries in MMH where the semantics of the queries are comparable yet the queries show disparately differing retrieval effectiveness. We are interested in disentangling a query representation into two independent representations one capturing query semantics and the other potentially capturing query difficulty. Ideally, once query representations are disentangled, the semantic component of the disentangled representations of similar queries would be similar. Additionally, one would expect that the difficulty component of the disentangled representations would enable us to discriminatively determine which query is more difficult for the retrieval method to satisfy, regardless of whether the compared queries have similar semantics or not. The benefit of this proposed query disentanglement approach is that it enables us to perform the pre-retrieval Query Performance Prediction (QPP) task in which the retrieval effectiveness of a query is determined prior to retrieval. Most QPP

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10…$15.00
https://doi.org/10.1145/3583780.3615189

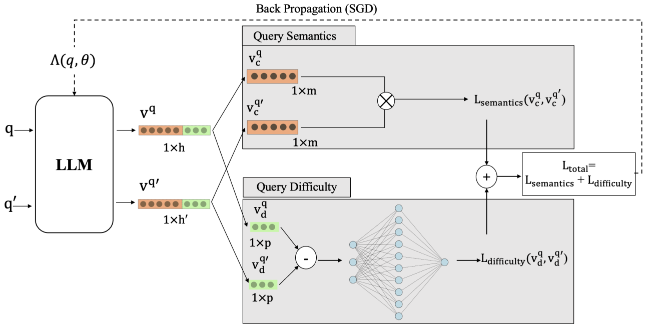

Figure 1: The overall architecture of our proposed method.

methods [2, 3, 6–8, 11, 14, 22, 23, 31, 36, 38] rely either on the statistical relation between the query terms and the document corpus, or on the relation between query terms within the geometric space produced by neural embedding models. However, in our work, we produce an explicit representation of query difficulty as a result of the disentanglement process, which can be directly used for performing QPP.

Through experiments on four widely used datasets, namely MS MARCO dev set, TREC DL 2019, TREC DL 2020, as well as TREC DL-Hard, and based on three evaluation metrics including Kendall and Spearman Correlations as well as the scaled Mean Absolute Ranking Error (sMARE) metric [13], we show that the proposed disentanglement approach is able to effectively capture query difficulty in isolation. Furthemore, and in order to show that the proposed approach is also able to isolate query semantics, we evaluate how well the disentangled representations are able to compute the association between semantically similar query pairs released by Nogueira and Lin [33]. We show that the disentangled representations of content are more effective in capturing query semantics compared to various state of the art large language models.

2 PROPOSED METHOD

Our method aims to disentangle query representations into distinct representations for query semantics and difficulty.

2.1 Representation Disentanglement

Let us assume a set of query pairs $Q_P = (q_i, q'_i)|i = 1, ..., N$, where N is the total number of query pairs. In each pair, the two queries can be defined in relation to each other as follows:

- *Query Semantics:* The queries in each pair could either be discussing different semantics or maybe the formulation of the same query but using different terms. For example, the two queries 'what does dse stand for' and 'what is display screen equipment' are both semantically similar and convey the same intent. In contrast, for another pair of queries such as 'average dishwasher life expectancy' and 'life expectancy in canada', the intent is completely different and their semantics are not comparable. As such, for any given pair, if the two queries share the same semantics, we label them as being similar ($l = 1$), otherwise, we label them as being dissimilar ($l = -1$).
- *Query Difficulty:* Regardless of query semantics, a pair of queries can be compared with each other based on their retrieval effectiveness. Queries with lower retrieval effectiveness are more difficult queries to satisfy. We define function ψ, which measures the performance of a query q over a collection D with a ranking model M. Therefore, q could be considered as a measure for difficulty of a query; The lower the performance of the query q as measured by $\psi(q)$, the higher its level of difficulty. For example, consider the previous example pairs: 'what does dse stand for' and 'what is display screen equipment'. The first query has an average precision of 0.05 while the second has an average precision of 1. Regardless of whether a pair of two queries are semantically comparable or not, we assign the pair a label of 1 if the first query q is more difficult than q', and 0 otherwise.

For this purpose of representation disentanglement, we further extend the Q_P set into Q as follows:

$$Q = \{(q_i, q'_i, l, y)|l \in \{-1, 1\}, y \in \{0, 1\}, \psi(q) \neq \psi(q'), i = 1, ..., N\} \quad (1)$$

We note that $\psi(q) \neq \psi(q')$ indicates that query pairs in Q do not have the same retrieval effectiveness.

2.2 Model Architecture

The architecture of our proposed model is depicted in Figure 1. The model focuses on decomposing the query representation into non-overlapping sub-representations of query semantics and difficulty.

Query Representation Encoding. Our architecture initially encodes each query q into latent space through $\Lambda(q; \Theta)$ where the vector representation v of query q is denoted as $v^q = \Lambda(q; \Theta)$. Let us assume the vector representation v is an h-dimensional vector. As illustrated in Figure 1, vector v would ideally be decomposed into two non-overlapping vectors v_c^q and v_d^q with m, and p dimensions, respectively, such that $h = m + p$. The idea is to consider the first part of the vector v_c^q to represent query semantics and the second part v_d^q as the query difficulty. Given the set of query pairs defined in Equation 1, the vector representations of each of the queries q, and q' can be calculated as $v^q = \Lambda(q; \Theta)$ and $v^{q'} = \Lambda(q'; \Theta)$.

Query Semantics. To disentangle query semantics, we train our architecture through a Cosine Embedding Loss function through which query pairs with similar and dissimilar semantics will have a 1 and -1 label respectively. More specifically, the architecture would only consider a subset of the original query representation for capturing query semantics, namely v_c^q and $v_c^{q'}$. Intuitively speaking, the training process would need to accumulate all query semantic information from the original query representation and squash them into a subset of the representation. The specific loss function for ensuring query semantics are captured in v_c^q and $v_c^{q'}$ can be defined as follows:

$$L_{Semantics} = \frac{1}{N} \sum_{i=1}^{N} [\frac{1+l}{2}(1 - \cos(v_c^{q_i}, v_c^{q'_i})) + \frac{1-l}{2} max(0, \cos(v_c^{q_i}, v_c^{q'_i}))] \quad (2)$$

where $\cos(v_c^q, v_c^{q'})$ is the cosine similarity between the two vectors v_c^q,$v_c^{q'}$. The embedding network Λ is fine-tuned such that it learns a compact yet rich representation for the semantics of each query. Representations learnt based on the embedding network will place queries with similar semantics closer to each other in the embedding space and distant semantically dissimilar queries from each other.

Query Difficulty. We tend to ensure that the disentangled representation is able to distinctly capture query difficulty in isolation from query semantics. As such, the remainder of the query representation in each query, namely v_d^q would need to capture the difficulty of each query based on $\psi(q)$ and in relation to another disentangled query representation such as $v_d^{q'}$. In order to train a model to be able to predict whether q is more difficult than q' or not solely based on their disentangled representation, we define a classification task. We note that the classification task needs to be cognizant of query pair ordering. More specifically, depending on the order in which we place the pair of queries in relation to each other, the difficulty relation would be reversed. In order to capture both the query difficulty relation and ordering, we work with the difference between the vector representation of the two disentangled queries in the classification task. We feed the difference of the two vectors $v_d^q - v_d^{q'}$ to a prediction network $\Pi(x;\Phi)$ and train it to classify this vector based on the labels $y \in \{0,1\}$. The loss function for such a network is defined as Binary Cross Entropy Loss as follows:

$$L_{Difficulty} = \frac{1}{N}\sum_{i=1}^{N}[-ylog(\Pi(v_d^{q_i} - v_d^{q'_i})) + (1-y)log(1-\Pi(v_d^{q_i} - v_d^{q'_i}))] \quad (3)$$

Overall Disentanglement Loss. The overall loss function of the network is defined as a linear interpolation of the two losses:

$$L_{Total} = L_{Semantics} + L_{Difficulty}.$$

3 EXPERIMENTS

Experimental Setup. We utilized the model in [16] as our transformer-based encoder model to map the queries into a 768-dimensional space. These representations were subsequently disentangled into query semantics and query difficulty vectors, which were chosen to be 500 and 268 dimensional, respectively. The fully connected layer had a size of 512. We found this architecture most robust to model performance during validation. We split our dataset to 80% train and 20% validation. We used the stochastic gradient descent optimization approach with the learning rate set to 10^{-3} and trained our model for 5 epochs with Batch size of 32.

Datasets. We constructed a dataset consisting of query pairs alongside their corresponding Mean Average Precision at 1000 performance values. We utilized query pairs sourced from the MMH dataset [1], which encompasses over 400,000 query pairs and their relative performance in relation to each other. We have also incorporated the MS MARCO [32] V1 train queries and their performance metrics. This integration resulted in a final dataset comprising 420,000 query pairs with similar content (labeled as 1) and an equal number of pairs with different content (labeled as −1), along with their respective performance values. To assess the performance of our approach under varying distributions of −1 and 1 labeled pairs, we conducted validation experiments using datasets with different label proportions. Notably, the results revealed that a dataset distributed evenly with 50% of each label type yielded the best performance overall. Our dataset is available to download on our GitHub repository (https://github.com/sara-salamat/query-disentanglement).

Evaluation. We conducted experiments on MS MARCO v1 passage collection and four of its accompanying query sets[32]. We compare the performance of our proposed method as well as the baselines on MS MARCO small development set which comprises 6,980 queries, most of which have only one relevant judgement. We also consider TREC DL 2019 [10] (43 queries), TREC DL 2020 [9] (53 queries) and DL-Hard [29] (50 queries) which all include comprehensive judgements on a non-binary graded scale. By testing our proposed method on these different query sets, we can compare the robustness of our approach as well as the baselines in terms of query size and the number of relevant documents per query. We evaluate the QPP effectiveness by two widely used evaluation strategies. First, we compute the Kendall and Spearman correlation between the predicted performance given by our method and the actual performance of the queries over the BM25 implemented by Anserini [42] quantified by the official evaluation metric for each query set, i.e., MRR@10 for MS MARCO dev set and nDCG@10 for the other three datasets. A higher correlation value indicates a more accurate prediction of query performance. In addition, we adopted the scaled Mean Absolute Relative Error (sMARE) evaluation metric which is a recently proposed metric for assessing the performance of QPP methods [13]. It measures the accuracy of predicted scores or rankings compared to the ground truth scores or rankings. sMARE has shown to be useful in quantifying the performance of QPP methods, providing a measure of how closely the predicted scores or rankings align with the ground truth. A lower sMARE value indicates better prediction accuracy.

Table 1: The List of QPP Baselines.

Category	Methods and Citation
Term Importance	IDF , ICTF [24]
Specificity	SCS [15], IEF [6], CC and DC [7]
Similarity	SCQ [44]
Term Relatedness	PMI [14]
Coherency	VAR [44]

To measure the impact of our method on capturing query semantics, we use four pre-trained models, namely RoBERTa [27], MPNet[39], MiniLM [40], BERT [12], and perform the disentanglement process on each of them. We then compare how well the disentangled portion of the query representation that captures query semantics is able to grasp query semantics compared to the original query representation prior to the disentanglement process. For this purpose, we adopt a subset of the dataset released by Nogueira and Lin [33] which consists of a set of original queries and 25 semantically similar yet alternatively expressed queries for each of the original queries. We compute the similarity between the original query and its 25 alternatives using the original query representation as well as our disentangled representation based on the Cosine Similarity function. An accurate representation would be one that would show higher semantic similarity when comparing these pairs.

Baselines For the QPP Task. We have adopted widely used pre-retrieval query performance prediction methods that have demonstrated promising performance on various popular corpora and query sets [6, 8]. The list of these baselines are included in Table 1.

Hyperparameter Setting. All baselines are reported based on their best-performing hyperparameters as reported in their original paper.

Table 2: Performance Comparison. *Italic* values indicate *not* statistically significant correlation with p-value of 0.05.

	MS MARCO Dev set			TREC DL 2019			TREC DL 2020			TREC DL Hard		
	Kendall τ	Spearman	sMARE	Kendall τ	Spearman	sMARE	Kendall τ	Spearman	sMARE	Kendall τ	Spearman	sMARE
SCQ	*0.011*	*0.014*	0.334	0.116	0.162	0.387	0.076	0.132	0.365	0.127	0.179	0.369
SCS	*0.037*	0.049	0.333	0.194	0.287	0.316	0.272	**0.397**	0.333	0.106	0.140	0.326
VAR	0.062	0.083	0.333	0.107	0.152	0.290	0.059	0.077	0.318	*0.016*	*0.035*	0.349
PMI	*0.017*	*0.023*	0.323	0.009	*0.017*	0.341	0.040	0.056	0.344	*0.022*	*0.031*	0.349
IDF	0.116	0.154	0.330	0.158	0.245	0.321	0.245	0.353	0.374	0.111	0.125	**0.255**
ICTF	0.114	0.152	0.330	0.153	0.240	0.360	0.345	0.330	0.330	0.107	0.115	0.314
CC	0.065	0.085	0.333	0.099	0.055	0.319	0.106	*0.026*	0.290	0.103	0.141	0.310
DC	0.107	0.144	0.333	0.095	0.053	0.293	0.091	*0.035*	0.327	0.123	0.165	0.335
IEF	0.094	0.104	0.330	0187	0.166	0.387	0.064	0.081	0.334	0.140	0.191	0.377
Ours	**0.24**	**0.359**	**0.259**	**0.2**	**0.3**	**0.273**	**0.274**	0.385	**0.248**	**0.171**	**0.257**	0.271

3.1 Results and Findings

Query Difficulty Prediction. Table 1 shows a comparison of our approach and the baselines. Based on the results in this table, we make several observations: **(1)** Our proposed method consistently achieves a statistically significant correlation with a $p{-}value < 0.05$, outperforming all the baselines across all datasets in terms of reported rank-based correlations. Our proposed method exhibits a lower sMARE value compared to the baselines. This is a strong advantage of our work since sMARE measures the discrepancy between the rankings of queries in actual and predicted ranks. Thus, a lower sMARE indicates superior performance. **(2)** Among these baselines, the ones based on term-importance, specifically IDF and ICTF, demonstrate superior performance compared to the others. However, when gauged on the sMARE metric, their efficacy is not as impressive. **(3)** Generally, the performance of the methods is significantly higher on the TREC DL datasets compared to the MS MARCO dev set. This may be due to the more comprehensive relevance judgements of the TREC DL 2019, 2020, and Hard datasets. The availability of more relevant judgments per query allows for a more precise performance evaluation, thus, more reliable performance prediction on these datasets. In contrast, the MS MARCO dev set typically contains an average one relevance judgment per query. Previous studies have indicated that incomplete judgements in such cases could lead to less accurate performance [4]. **(4)** Our proposed method showed better performance than all the baselines on the four corpora in terms of all three evaluation metrics. In TREC-DL 2020, SCS achieved a Spearman ρ of 0.397 while our method yielded a correlation of 0.385 (second best). On TREC DL-Hard, we obtained an sMARE of 0.271 (runner up), while IDF managed a lower sMARE of 0.255. However, upon a more in-depth examination of these cases, we found that these differences were not statistically significant based on a paired t-test with a p-value of less than 0.05. Furthermore, neither SCS nor IDF demonstrated consistent effectiveness across the four different query sets. In contrast, our method achieved 0.24 in terms of Kendall τ and 0.359 in terms of Spearman ρ, representing a substantial boost in performance on the MS MARCO dev set. Overall, our method exhibits the highest consistency and overall performance, indicating its robustness in relation to varying query subsets and evaluation strategies.

Query Semantics Performance. We assess the effectiveness of our disentanglement process for capturing query semantics by comparing query representations before and after the disentanglement process for various models including RoBERTa [27], MPNet [39], MiniLM [40], and BERT [12]. To ascertain how well each disentangled representation encapsulates query semantics, we visualize the distributions of similarities between the query pairs in the dataset proposed by Nogueira and Lin [33]. Given this dataset consists of matches between original queries with another 25 corresponding queries, it is possible to compute the similarity between these pairs. An effective query representation that has accurately captured query semantics would embed each of the query pairs closer to each other within the embedding space. Figure 2 shows the histogram of similarities between 5,000 randomly chosen query pairs from the Nogueira and Lin dataset [33]. A histogram skewed towards the right shows higher similarity values between query pairs indicating that the model has been able to position similar queries in closer proximity (a desirable outcome). As depicted in the figure, the disentanglement process has made all four language models to show higher skewness to the right showing that our proposed approach will enable a more accurate capture of query semantics.

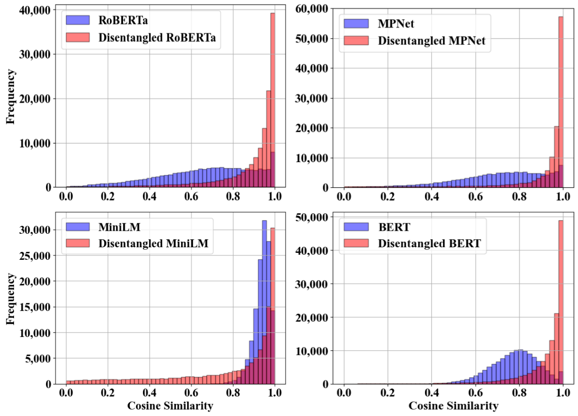

Figure 2: The semantic similarity histograms.

4 CONCLUDING REMARKS

The objective of our work in this paper has been to design a neural disentanglement method that is able to capture and isolate the representation of query semantics from that of query difficulty. We achieve this objective by fine-tuning portions of query representations to capture the similarity and dissimilarity between relevant and non-relevant queries, respectively. Furthermore, we fine-tune a non-overlapping portion of the query representation to effectively capture the retrieval effectiveness of queries. We have shown that our approach has been able to show strong performance on the QPP task compared to the state of the art and also show better performance on query similarity calculation compared to various language models.

REFERENCES

[1] Negar Arabzadeh, Amin Bigdeli, Shirin Seyedsalehi, Morteza Zihayat, and Ebrahim Bagheri. 2021. Matches Made in Heaven: Toolkit and Large-Scale Datasets for Supervised Query Reformulation. In *Proceedings of the 30th ACM Int'l Conf. on Information & Knowledge Management*. 4417–4425.

[2] Negar Arabzadeh, Amin Bigdeli, Morteza Zihayat, and Ebrahim Bagheri. 2021. Query Performance Prediction Through Retrieval Coherency. In *Advances in Information Retrieval: 43rd European Conf. on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part II 43*. Springer, 193–200.

[3] Negar Arabzadeh, Maryam Khodabakhsh, and Ebrahim Bagheri. 2021. BERT-QPP: contextualized pre-trained transformers for query performance prediction. In *Proceedings of the 30th ACM Int'l Conference on Information & Knowledge Management*. 2857–2861.

[4] Negar Arabzadeh, Alexandra Vtyurina, Xinyi Yan, and Charles LA Clarke. 2022. Shallow pooling for sparse labels. *Information Retrieval Journal* 25, 4 (2022), 365–385.

[5] Negar Arabzadeh, Xinyi Yan, and Charles LA Clarke. 2021. Predicting efficiency/effectiveness trade-offs for Dense vs. Sparse retrieval strategy selection. In *Proceedings of the 30th ACM Int'l Conference on Information & Knowledge Management*. 2862–2866.

[6] Negar Arabzadeh, Fattane Zarrinkalam, Jelena Jovanovic, Feras Al-Obeidat, and Ebrahim Bagheri. 2020. Neural embedding-based specificity metrics for pre-retrieval query performance prediction. *Information Processing & Management* 57, 4 (2020), 102248.

[7] Negar Arabzadeh, Fattane Zarrinkalam, Jelena Jovanovic, and Ebrahim Bagheri. 2020. Neural embedding-based metrics for pre-retrieval query performance prediction. In *Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part II 42*. Springer, 78–85.

[8] David Carmel and Elad Yom-Tov. 2010. Estimating the query difficulty for information retrieval. *Synthesis Lectures on Information Concepts, Retrieval, and Services* 2, 1 (2010), 1–89.

[9] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. Overview of the TREC 2020 deep learning track. *CoRR* abs/2102.07662 (2021). arXiv:2102.07662 https://arxiv.org/abs/2102.07662

[10] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees. 2020. Overview of the TREC 2019 deep learning track. *arXiv preprint arXiv:2003.07820* (2020).

[11] Steve Cronen-Townsend, Yun Zhou, and W Bruce Croft. 2002. Predicting query performance. In *Proceedings of the 25th annual Int'l ACM SIGIR Conf. on Research and development in information retrieval*. 299–306.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[13] Guglielmo Faggioli, Oleg Zendel, J Shane Culpepper, Nicola Ferro, and Falk Scholer. 2022. sMARE: a new paradigm to evaluate and understand query performance prediction methods. *Information Retrieval Journal* 25, 2 (2022), 94–122.

[14] Claudia Hauff. 2010. Predicting the effectiveness of queries and retrieval systems. In *SIGIR Forum*, Vol. 44. 88.

[15] Ben He and Iadh Ounis. 2004. Inferring Query Performance Using Pre-retrieval Predictors.. In *String Processing and Information Retrieval, 11th Int'l Conf., SPIRE 2004, Padova, Italy, October 5-8, 2004, Proceedings*. 43–54. https://doi.org/10.1007/978-3-540-30213-1_5

[16] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654* (2020).

[17] Jie Hu, Liujuan Cao, Tong Tong, Qixiang Ye, Shengchuan Zhang, Ke Li, Feiyue Huang, Ling Shao, and Rongrong Ji. 2021. Architecture disentanglement for deep neural networks. In *Proceedings of the IEEE/CVF Int'l Conference on Computer Vision*. 672–681.

[18] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward Controlled Generation of Text. In *Proceedings of the 34th Int'l Conf. on Machine Learning - Vol. 70 (ICML'17)*. JMLR.org, 1587–1596.

[19] Zhenya Huang, Xin Lin, Hao Wang, Qi Liu, Enhong Chen, Jianhui Ma, Yu Su, and Wei Tong. 2021. Disenqnet: Disentangled representation learning for educational questions. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 696–704.

[20] Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, and Mingli Song. 2019. Neural style transfer: A review. *IEEE transactions on visualization and computer graphics* 26, 11 (2019), 3365–3385.

[21] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906* (2020).

[22] Maryam Khodabakhsh and Ebrahim Bagheri. 2021. Semantics-enabled query performance prediction for ad hoc table retrieval. *Information Processing & Management* 58, 1 (2021), 102399.

[23] Heejin Kim and Kyung-Ah Sohn. 2020. How Positive Are You: Text Style Transfer using Adaptive Style Embedding. In *Proceedings of the 28th Int'l Conf. on Computational Linguistics*. Int'l Committee on Computational Linguistics, Barcelona, Spain (Online), 2115–2125. https://doi.org/10.18653/v1/2020.coling-main.191

[24] K. L. Kwok. 1996. A New Method of Weighting Query Terms for Ad-Hoc Retrieval. In *Proceedings of the 19th Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval, SIGIR'96, August 18-22, 1996, Zurich, Switzerland (Special Issue of the SIGIR Forum)*. 187–195. https://doi.org/10.1145/243199.243266

[25] Jongpil Lee, Nicholas J Bryan, Justin Salamon, Zeyu Jin, and Juhan Nam. 2020. Metric learning vs classification for disentangled music representation learning. *arXiv preprint arXiv:2008.03729* (2020).

[26] Hang Li and Zhengdong Lu. 2016. Deep learning for information retrieval. In *Proceedings of the 39th Int'l ACM SIGIR conference on Research and Development in Information Retrieval*. 1203–1206.

[27] 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* (2019).

[28] Marc Moreno Lopez and Jugal Kalita. 2017. Deep Learning applied to NLP. *arXiv preprint arXiv:1703.03091* (2017).

[29] Iain Mackie, Jeffrey Dalton, and Andrew Yates. 2021. How Deep is your Learning: the DL-HARD Annotated Deep Learning Dataset. In *Proceedings of the 44th Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval*.

[30] Shikib Mehri and Giuseppe Carenini. 2017. Chat disentanglement: Identifying semantic reply relationships with random forests and recurrent neural networks. In *Proceedings of the Eighth Int'l Joint Conference on Natural Language Processing (Vol. 1: Long Papers)*. 615–623.

[31] Chuan Meng, Negar Arabzadeh, Mohammad Aliannejadi, and Maarten de Rijke. 2023. Query Performance Prediction: From Ad-hoc to Conversational Search. *arXiv preprint arXiv:2305.10923* (2023).

[32] 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. *choice* 2640 (2016), 660.

[33] Rodrigo Nogueira, Jimmy Lin, and AI Epistemic. 2019. From doc2query to docTTTTTquery. *Online preprint* 6 (2019).

[34] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Jingfang Xu, and Xueqi Cheng. 2017. Deeprank: A new deep architecture for relevance ranking in information retrieval. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 257–266.

[35] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

[36] Dwaipayan Roy, Debasis Ganguly, Mandar Mitra, and Gareth JF Jones. 2019. Estimating gaussian mixture models in the local neighbourhood of embedded word vectors for query performance prediction. *Information processing & management* 56, 3 (2019), 1026–1045.

[37] Yashvardhan Sharma and Sahil Gupta. 2018. Deep learning approaches for question answering system. *Procedia computer science* 132 (2018), 785–794.

[38] Anna Shtok, Oren Kurland, David Carmel, Fiana Raiber, and Gad Markovits. 2012. Predicting query performance by query-drift estimation. *ACM Transactions on Information Systems (TOIS)* 30, 2 (2012), 1–35.

[39] 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 (2020), 16857–16867.

[40] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems* 33 (2020), 5776–5788.

[41] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808* (2020).

[42] Peilin Yang, Hui Fang, and Jimmy Lin. 2017. Anserini: Enabling the use of lucene for information retrieval research. In *Proceedings of the 40th Int'l ACM SIGIR Conf. on research and development in information retrieval*. 1253–1256.

[43] Zihan Ye, Fuyuan Hu, Fan Lyu, Linyan Li, and Kaizhu Huang. 2021. Disentangling semantic-to-visual confusion for zero-shot learning. *IEEE Transactions on Multimedia* 24 (2021), 2828–2840.

[44] Ying Zhao, Falk Scholer, and Yohannes Tsegay. 2008. Effective Pre-retrieval Query Performance Prediction Using Similarity and Variability Evidence. In *Advances in Information Retrieval , 30th European Conf. on IR Research, ECIR 2008, Glasgow, UK, March 30-April 3, 2008. Proceedings*. 52–64. https://doi.org/10.1007/978-3-540-78646-7_8

[45] Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guangjing Wang, Kai Zhang, Cheng Ji, Qiben Yan, Lifang He, et al. 2023. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *arXiv preprint arXiv:2302.09419* (2023).

[46] Anna Zhu, Zhanhui Yin, Brian Kenji Iwana, Xinyu Zhou, and Shengwu Xiong. 2022. Text Style Transfer Based on Multi-Factor Disentanglement and Mixture. In *Proceedings of the 30th ACM Int'l Conf. on Multimedia (MM '22)*. ACM, New York, NY, USA, 2430–2440. https://doi.org/10.1145/3503161.3548239