Dealing with Typos for BERT-based Passage Retrieval and Ranking

Shengyao Zhuang
The University of Queensland
Brisbane, QLD, Australia
s.zhuang@uq.edu.au

Guido Zuccon
The University of Queensland
Brisbane, QLD, Australia
g.zuccon@uq.edu.au

Abstract

Passage retrieval and ranking is a key task in open-domain question answering and information retrieval. Current effective approaches mostly rely on pre-trained deep language model-based retrievers and rankers. These methods have been shown to effectively model the semantic matching between queries and passages, also in presence of keyword mismatch, i.e. passages that are relevant to a query but do not contain important query keywords.

In this paper we consider the Dense Retriever (DR), a passage retrieval method, and the BERT re-ranker, a popular passage re-ranking method. In this context, we formally investigate how these models respond and adapt to a specific type of keyword mismatch – that caused by *keyword typos* occurring in queries. Through empirical investigation, we find that typos can lead to a significant drop in retrieval and ranking effectiveness. We then propose a simple typos-aware training framework for DR and BERT re-ranker to address this issue. Our experimental results on the MS MARCO passage ranking dataset show that, with our proposed typos-aware training, DR and BERT re-ranker can become robust to typos in queries, resulting in significantly improved effectiveness compared to models trained without appropriately accounting for typos.

1 Introduction

Passage ranking is a core task for many information retrieval related applications. In the context of conversational search and question answering, for example, passage ranking is often the first step in the system's pipeline: thus the quality of the ranking results will affect the effectivenesses of the downstream tasks. Traditional passage ranking models, such TF-IDF and BM25, use exact keyword matching signals, where a retrieved passage must contain at least one of the query's keywords. This mechanism however limits the capability of these models to retrieve passages that are semantically relevant but use different keywords: this is the well-known vocabulary mismatch problem.

Recent advances in NLP have seen the introduction of deep language models (Devlin et al., 2018; Brown et al., 2020; Raffel et al., 2020); BERT (Devlin et al., 2018) in particular has shown generalised promise in language understanding tasks. BERT adopts the transformer encoder (Vaswani et al., 2017) as model architecture and uses WordPiece token embeddings (Wu et al., 2016) as model inputs. This design allows BERT to deal with the vocabulary mismatch problem. Hence practitioners have turned to design BERT-based passage ranking models (Lin et al., 2020).

Two main directions have been adopted to exploit BERT for effective passage ranking:

- **Dense Retriever (DR)** (Zhan et al., 2020; Xiong et al., 2020; Gao et al., 2021b; Khattab and Zaharia, 2020; Karpukhin et al., 2020; Ding et al., 2020; Luan et al., 2020): queries and passages are separately encoded into low-dimensional dense representations with BERT. At indexing time, passage representations are computed and then stored in the index. At query time, a single query encoder inference is needed to obtain the query representation; then passage relevance scores are estimated by computing the similarity between the query and passages' representations.
- **BERT re-ranker**, a.k.a. monoBERT (Nogueira and Cho, 2019; Dai and Callan, 2019; Gao et al., 2021a): the ranking task is modelled as a classification task that builds upon the BERT model. The input to BERT is a $< query, passage >$ pair and the relevance score can be computed by a linear layer on the $< CLS >$ token embedding, or the query likelihood estimated by the BERT model (Zhuang and Zuccon, 2021b). A key drawback of BERT re-ranker models is that multiple inferences are required at query time: this is a computationally expensive process

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 2836–2842
November 7–11, 2021. ©2021 Association for Computational Linguistics

which results in high query latency (Zhuang and Zuccon, 2021a; MacAvaney et al., 2020; Hofstätter et al., 2020). Thus the use of these methods is confined to second stage re-ranking.

In principle, these methods are not affected by keyword mismatch because they use the latent embedding space to estimate the relevance of a query to a passage. This is supported by recent work that has shown the DR and BERT re-ranker provide better semantic matching (Zhan et al., 2020; MacAvaney et al., 2020; Formal et al., 2021).

In this paper we investigate the impact of a specific type of keyword mismatch: that caused by the presence of typos in the query. Traditional exact keyword matching methods perform badly on queries that contain typos. Extra query processing steps, such as spelling correction, are required for these methods to be tolerant to typos in queries (Martins and Silva, 2004). On the other hand, it is expected that BERT-based models can handle typos occurring in queries well. This is because BERT uses the WordPiece algorithm which splits a keyword that does not match an entry in the BERT vocabulary (typos are likely to not be present in this vocabulary) into character-level sub-tokens. This can be used to produce embeddings for out-of-vocabulary keywords, which are then passed as input to the BERT encoder. However, this intuition has never been tested before, and the capability of BERT-based passage ranking models to deal with typos in queries has not been quantified.

To address this gap, we first formally investigate how the BERT-based DR and re-ranker respond and adapt to queries that contain typos. Specifically, we use different typo generators to produce typos for queries; we then compare the effectiveness of the rankers when using queries with typos vs. without typos. Interestingly, we find that these models fail to handle queries with typos – typos can lead to a significant drop in effectiveness for both DR and BERT re-ranker. In order to solve this issue and obtain typo-robust ranking models, we then propose a simple typos-aware training strategy, in which queries with typos are produced and used also for training. Our experimental results on the MS MARCO passage ranking dataset show that, with our typos-aware training, DR and BERT re-ranker can become robust to typos in queries, without loss in effectiveness for queries without typos.

2 Methodology

With respect to BERT-based models for passage retrieval and typos in queries, we investigate the following research questions:

- **RQ1:** What is the impact of typos in queries on BERT-based DR and re-ranker effectiveness?
- **RQ2:** Do different typo types affect the effectiveness of the BERT-based methods differently?
- **RQ3:** Does the proposed typos-aware training improve the effectiveness of the BERT-based methods on queries with typos? Does it hurt their effectiveness on queries without typos?

2.1 Synthetic Typo Generation

To answer our research questions, a reasonably large set of queries with different types of typos is needed. As there is no available dataset for passage retrieval with labels that indicate the presence of typos in queries, we set off to create one such dataset. For this we augmented the MS MARCO passage retrieval dataset. (Manual inspection of this dataset did not reveal a considerable amount of queries with typos; the dataset curators likely did manually remove most typos). For augmentation, we synthetically generated typos from the original queries in the dataset, so that we could carefully control the number and types of typos.

For generating typos, we used the following operations that give rise to typos that often occur in real-world queries (Hagen et al., 2017):

- **Random character Insertion (RandInsert):** Inserts a random letter into a random word, e.g., “search typo” -> “search tyapo”.
- **Random character deletion (RandDelete):** Deletes a random character of a random word, such as “search typo” -> “search tpo”.
- **Random character substitution (RandSub):** Randomly replaces a character of a random word with a random letter, e.g., “search typo” -> “search type”.
- **Swap neighbor character (SwapNeighbor):** Randomly swaps a character with one of its neighbor characters, e.g., “search typo” -> “search tyop”.
- **Swap adjacent keyboard character (SwapAdjacent)**: Randomly swaps a character with one of its adjacent letter on the keyboard[1], e.g., “search typo” -> “search typi”.

[1] e.g., on a *QWERTY* keyboard, the list of adjacent characters for character ‘s’ is [‘q’, ‘w’,‘e’, ‘a’, ‘d’, ‘z’, ‘x’, ‘c’].

Since queries in MS MARCO are relatively short ($\approx$ 6 keywords on average) (Nguyen et al., 2016), when generating typos for queries, we only consider keywords that have more than 3 characters and only randomly modify one keyword per query. We use open-source tool kits TextAttack (Morris et al., 2020) to implement these typo generators.

2.2 Typos-aware Training

To deal with queries with typos we propose to consider such queries also during the training phase of DR and BERT re-ranker: we call this typos-aware training. Specifically, for each original query that appears during the training phase, we draw an unbiased coin. If the result is head, we leave the query unchanged and use it for training. If it is tail (50% chances) we inject a typo in the query by uniformly sampling one of the considered typos generators (Section 2.1), and use the modified query for training. By doing so, at training time, the BERT-based methods will observe both the original, typos-free, queries and queries with different types of typos. Thus, in order to reduce the training loss, we force the methods to learn to be invariant to different types of typos.

Our typos-aware training can be considered a data augmentation approach, with small perturbations to some training queries: these do not change the underlying intent of the query or the relevance of the target passage. Data augmentation has been shown effective for a range of deep learning tasks, including computer vision (He et al., 2020; Chen et al., 2020; Grill et al., 2020) and NLP (Zhang et al., 2015; Wei and Zou, 2019; Xie et al., 2020; Jiao et al., 2020); however, the impact of data augmentation on ad-hoc retrieval remains to be studied.

3 Experimental Settings

3.1 Dataset and Evaluation Measures

For evaluation, we use the MS MARCO passage ranking dataset (Nguyen et al., 2016), which consists of 8.8M passages, $\approx$503K training queries and 6,980 dev queries. For typos-aware training, we modify training queries with a 50% chance. For dev queries, we experiment with both the original queries and with queries modified to contain typos. We produce typos for all queries in the dev set using the strategies in Section 2.1; we also consider the average effectiveness across typos queries.

We use the official metric MRR@10 to evaluate the ranking effectiveness of both DR and BERT re-ranker. We use the BERT re-ranker as a second stage ranker, on top of the initial rankings provided by DR. This is unlike previous work (Lin et al., 2020), in which the BERT re-ranker is typically used on top of BM25. Our setting is motivated by the fact that BM25 would fail to retrieve the relevant target passages for queries that contain typos. Because of this, we also report Recall@1000 (labelled Recall) for DR, as this forms the basis of the first stage of retrieval and the number of retrieved relevant passages affects the effectiveness of the BERT re-ranker. Recall for BERT re-ranker is thus the same as that of DR, and is not reported.

3.2 DR Training Details

We follow Zhan et al. (2020) when training the DR. We adopt the BERT-Siamese architecture in which the query encoder and passage encoder share the BERT model parameters. This architecture has been used consistently in many recent approaches (Luan et al., 2020; Xiong et al., 2020). We use pairwise hinge loss with the "Train Triples" data provided in MS MARCO to fine-tune the “bert-base-uncased" model from the Huggingface library (Wolf et al., 2020). We use the ADAM optimizer, learning rate of 3e-6 with linear warm-up and decay scheduling. The model is trained on a single Tesla V100 GPU with a batch size of 26 and gradient accumulation step of 2 for 210K steps.

3.3 BERT re-ranker Training Details

To train the BERT re-ranker, we follow the training practice described by Nogueira and Cho (2019). We fine-tune a “bert-large-uncased" model with binary cross-entropy loss to perform binary classification on query-passage pairs. Negative pairs are randomly sampled from the top 1,000 passages retrieved by a trained DR model (without typos-aware training). We set the ratio of positive pairs to negative pairs to 1:4. We use the same optimizer and learning rate scheduling used for DR; the model is trained on two Tesla V100 GPUs with a batch size of 2 × 64 for 70K steps.

For both DR and BERT re-ranker typos-aware training, we use exactly the same setting used for the standard training described above.

4 Results

Empirical results are reported in Table 1. We note that BM25 if outperformed by both DR and BERT re-ranker across all settings, confirming the superi-

Table 1: MS MARCO passage ranking results. Row 2 reports results averaged across all typos queries; rows 3-7 results for each typos type (for each type, typos are injected in all dev queries). Percentage reductions are computed w.r.t. the original queries; bold represents best performance across training methods for each of DR and BERT re-ranker. Statistical significant gains (two-tailed paired t-test with Bonferroni correction, $p < 0.01$) obtained by models with typos-aware training over the models with standard training (std.) are indicated by †.

	BM25		**DR (std.)**		**DR (typos-aware)**		**Re-ranker (std.)**	**Re-ranker (typos-aware)**
Typo type	MRR@10	Recall	MRR@10	Recall	MRR@10	Recall	MRR@10	MRR@10
original	.187	.857	.296	.940	**.300**	.940	**.379**	.374
w. typos (avg)	.120(−35.8%)	.696(−18.6%)	.141(−52.3%)	.712(−24.3%)	**.219**†(−27.0%)	**.857**†(−8.8%)	.250(−34.0%)	**.289**†(−22.7%)
RandInsert	.125(−33.1%)	.693(−18.9%)	.140(−52.7%)	.711(−24.4%)	**.225**†(−25.0%)	**.862**†(−8.3%)	.257(−32.2%)	**.297**†(−20.6%)
RandDelete	.118(−36.9%)	.693(−18.9%)	.154(−47.9%)	.730(−22.3%)	**.217**†(−27.6%)	**.853**†(−9.3%)	.257(−32.2%)	**.288**†(−23.0%)
RandSub	.120(−35.8%)	.702(−17.9%)	.137(−53.7%)	.714(−24.0%)	**.220**†(−26.7%)	**.858**†(−8.7%)	.250(−34.0%)	**.291**†(−22.2%)
SwapNeighbor	.122(−34.7%)	.702(−17.9%)	.137(−53.7%)	.705(−25.0%)	**.217**†(−27.6%)	**.859**†(−8.6%)	.240(−36.7%)	**.284**†(−24.1%)
SwapAdjacent	.117(−37.4%)	.691(−19.1%)	.137(−53.7%)	.702(−25.3%)	**.214**†(−28.7%)	**.854**†(−9.1%)	.246(−35.1%)	**.286**†(−23.5%)

ority of BERT-based methods. For RQ1, we compare the results obtained on the original queries with those on queries with typos, when models are trained using the standard procedure. We observe statistically significant losses in effectiveness for both DR (on average MRR@10 drops 52.3% and Recall 24.3%) and BERT re-ranker (MRR@10 drops 34%). BERT re-ranker is performed on top of DR results, thus losses in Recall for DR are propagated to the BERT re-ranker. However, the effectiveness of DR on typos queries drops to about that of BM25, while BERT re-ranker stays superior.

In terms of the impact of different types of typos (RQ2), the results show that different typos have similar impact: they all hurt effectiveness heavily. DR appears however more tolerant to RandDelete typos (with a ≈ 5% smaller loss in MRR@10 than for other types of typos), while BERT re-ranker losses are generally uniform across typos types.

To answer RQ3, we compare the results of models produced with typos-aware training vs. with the standard training. Despite the typos-aware training, both DR and BERT re-ranker display significant losses in effectiveness when dealing with typos queries, compared to the original queries (Table 1). However, compared to the models with standard training, both methods are much more tolerant to all types of typos when typos-aware training is employed. In fact, losses in MRR@10 halve for DR (from 52.3% to 24.3%), and reduce by one third for BERT re-ranker (from 34% to 22.7%); all differences are statistically significant. Typos-aware training seems to impact queries with typos produced by RandomInsert more than those with other types of typos. We also note that effectiveness obtained by models with typos-aware training is not different from that with standard training if only queries without typos (original) are considered (minor differences are not statistically significant).

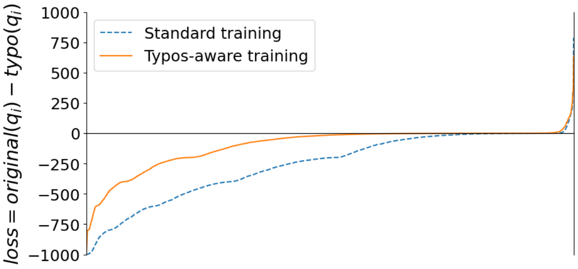

Figure 1: Loss in terms of the rank position of the first relevant passage retrieved by DR when ranking for typos queries, compared to original queries. Each point on the x-axis refers to a query; x-axis ordered by decreasing loss when standard training is used.

Figure 1 presents the rank loss obtained by DR when answering typos queries in place of the original queries (plot averaged across all types of typos; individual typos types show similar trends). A negative loss of n means when using typos queries, the first relevant document is retrieved n rank positions after that obtained when using the original queries. The figure shows that typos-aware training consistently provides smaller losses than the standard training. We also note there are few cases (≈ 300 queries) in which typos queries provide gains compared to the original query. Queries with large losses often have typos for keywords that are essential to determine the intent of the query. Typos queries that exhibit gains generally display typos on non-essential keywords, e.g., stopwords.

5 A Case Study

The results presented in the previous sections are conducted with synthetically generated typo queries. Accurate analysis of the MS MARCO dataset revels the presence of a very limit number of queries containing typos – these typos are legitimate errors made by the user issuing the query. For instance, the MS MARCO dev set contains the mistyped query – *"sydeny climate"* [2] (qid: 506025). Without typos-aware training, the considered DR cannot retrieve the relevant passage in the top 1,000 results. However, with typos-aware training, the DR is able to rank the relevant passage at rank 127 for this typo query. This suggests that DRs trained with our typos-aware training with synthetic typo generation may be able to generalize to real-world typo queries, aside from those synthetic (though realistic) typo queries we considered in our extensive empirical evaluation. In future work, we want to further test our proposed typos-aware training with more real-world typo queries by acquiring a real query log with typos and perform relevance annotations on the MS MARCO passage collection.

6 Conclusion

In this paper we studied the impact of typos in queries on popular BERT-based passage retrieval methods. We reported significant drops in effectiveness across different types of typos for both DR and BERT re-ranker: these methods are not tolerant to typos in queries when solely relying on the BERT encoder. We then proposed a typos-aware training strategy for DR and BERT re-ranker, which controls the exposure of the models to queries with typos during training. With our typos-aware training, both DR and BERT re-ranker showed to be much more tolerant to typos in queries. We believe our typos-aware training can be used (more extensively than in this paper) as a standard data augmentation step in the DR and BERT re-ranker's training loop since the computations for typos generation are very light and can provide extra gains on typos queries, without hurting effectiveness on queries without typos. Code, typos queries and results files at https://github.com/ielab/typos-aware-BERT.

Acknowledgements

Dr Guido Zuccon is the recipient of an Australian Research Council DECRA Research Fellowship (DE180101579). This research is partially funded by the Grain Research and Development Corporation project AgAsk (UOQ2003-009RTX).

[2] The correct spelling is "sydney climate".

References

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Zhuyun Dai and Jamie Callan. 2019. Deeper text understanding for ir with contextual neural language modeling. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 985–988.

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*.

Yingqi Qu Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2020. Rocketqa: An optimized training approach to dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2010.08191*.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. A white box analysis of colbert. In *The 43rd European Conference On Information Retrieval (ECIR)*.

Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021a. Rethink training of bert rerankers in multi-stage retrieval pipeline. In *The 43rd European Conference On Information Retrieval (ECIR)*.

Luyu Gao, Zhuyun Dai, Tongfei Chen, Zhen Fan, Benjamin Van Durme, and Jamie Callan. 2021b. Complementing lexical retrieval with semantic residual embedding. In *The 43rd European Conference On Information Retrieval (ECIR)*.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. 2020. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*.

Matthias Hagen, Martin Potthast, Marcel Gohsen, Anja Rathgeber, and Benno Stein. 2017. A large-scale query spelling correction corpus. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1261–1264.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738.

Sebastian Hofstätter, Markus Zlabinger, and Allan Hanbury. 2020. Interpretable & time-budget-constrained contextualization for re-ranking. In *ECAI 2020*, pages 513–520. IOS Press.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. Tinybert: Distilling bert for natural language understanding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 4163–4174.

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. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 39–48.

Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2020. Pretrained transformers for text ranking: Bert and beyond. *arXiv preprint arXiv:2010.06467*.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2020. Sparse, dense, and attentional representations for text retrieval. *arXiv preprint arXiv:2005.00181*.

Sean MacAvaney, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, Nazli Goharian, and Ophir Frieder. 2020. Expansion via prediction of importance with contextualization. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1573–1576.

Bruno Martins and Mário J Silva. 2004. Spelling correction for search engine queries. In *International Conference on Natural Language Processing (in Spain)*, pages 372–383. Springer.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126.

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*.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6383–6389.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33.

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*.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020. Repbert: Contextualized text embeddings for first-stage retrieval. *arXiv preprint arXiv:2006.15498*.

Xiang Zhang, Junbo Zhao, and Yann Lecun. 2015. Character-level convolutional networks for text classification. *Advances in Neural Information Processing Systems*, 2015:649–657.

Shengyao Zhuang and Guido Zuccon. 2021a. Fast passage re-ranking with contextualized exact term matching and efficient passage expansion. *arXiv preprint arXiv:2108.08513*.

Shengyao Zhuang and Guido Zuccon. 2021b. Tilde: Term independent likelihood model for passage re-ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.