Evaluating the Robustness of Retrieval Pipelines with Query Variation Generators

Penha, Gustavo; Câmara, Arthur; Hauff, Claudia

DOI
10.1007/978-3-030-99736-6_27

Publication date
2022

Document Version
Final published version

Published in
Advances in Information Retrieval - 44th European Conference on IR Research, ECIR 2022, Proceedings

Citation (APA)
Penha, G., Câmara, A., & Hauff, C. (2022). Evaluating the Robustness of Retrieval Pipelines with Query Variation Generators. In M. Hagen, S. Verberne, C. Macdonald, C. Seifert, K. Balog, K. Nørvåg, & V. Setty (Eds.), *Advances in Information Retrieval - 44th European Conference on IR Research, ECIR 2022, Proceedings* (pp. 397-412). (Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Vol. 13185 LNCS). Springer. https://doi.org/10.1007/978-3-030-99736-6_27

Evaluating the Robustness of Retrieval Pipelines with Query Variation Generators

Gustavo Penha(✉), Arthur Câmara, and Claudia Hauff

TU Delft, Delft, Netherlands
{g.penha-1,a.barbosacamara,c.hauff}@tudelft.nl

Abstract. Heavily pre-trained transformers for language modeling, such as BERT, have shown to be remarkably effective for Information Retrieval (IR) tasks, typically applied to re-rank the results of a first-stage retrieval model. IR benchmarks evaluate the effectiveness of retrieval pipelines based on the premise that a single query is used to instantiate the underlying information need. However, previous research has shown that (I) queries generated by users for a fixed information need are extremely variable and, in particular, (II) neural models are brittle and often make mistakes when tested with modified inputs. Motivated by those observations we aim to answer the following question: *how robust are retrieval pipelines with respect to different variations in queries that do not change the queries' semantics?* In order to obtain queries that are representative of users' querying variability, we first created a taxonomy based on the manual annotation of transformations occurring in a dataset (UQV100) of user-created query variations. For each syntax-changing category of our taxonomy, we employed different automatic methods that when applied to a query generate a query variation. Our experimental results across two datasets for two IR tasks reveal that retrieval pipelines are not robust to these query variations, with effectiveness drops of $\approx$ 20% on average. The code and datasets are available at https://github.com/Guzpenha/query_variation_generators.

1 Introduction

Heavily pre-trained transformers for language modeling such as BERT [17] have been shown to be remarkably effective for a wide range of IR tasks [40,43,55]. Commonly, IR benchmarks organized as part of TREC or other evaluation campaigns, evaluate the effectiveness of ranking models—neural or otherwise—based on small sets of topics and their corresponding relevance judgments. Importantly, each topic is typically represented by a single query[1]. However, previous research has shown that queries created by users given a fixed information need may vary widely [6,60]. In the UQV100 [5] dataset for instance, crowd workers on average

[1] While TREC topics usually consist of three parts (title, description and narrative), commonly only the TREC topic title is considered as query.

M. Hagen et al. (Eds.): ECIR 2022, LNCS 13185, pp. 397–412, 2022.
https://doi.org/10.1007/978-3-030-99736-6_27

created 57.7 unique queries for a given information need as instantiated via a backstory.

We thus argue that it is necessary to investigate the robustness of retrieval pipelines in light of *query variations* (i.e., different expressions of the same information need) that are *likely to occur in practice*. That different query variations lead to vastly different ranking qualities is anecdotally shown in Table 1 for a vanilla BERT model for ranking [40]. If, for example, the word order of the original query from TREC-DL-2019 *right pelvic pain causes* is changed to *causes pelvic pain right*, the retrieval effectiveness of the resulting ranking drops by 46%. Similarly, paraphrasing *define visceral* to *what is visceral* reduces the retrieval effectiveness by 38%.

Table 1. Examples of BERT effectiveness drops (nDCG@10 Δ) when we replace the original query from TREC-DL-2019 by an automatic (except for the first two lines that were produced manually) query variation. We focus here on transformations that change the query syntax , but not its semantics .

Original Query	**Query Variation**	**nDCG@10 Δ**
popular food in switzerland	popular food in zurich *gen./specialization*	
cost of interior concrete flooring	concrete flooring finishing *aspect change*	
what is theraderm used for	what is **thrraderm** used for *misspelling*	-1.00 (-100%)
anthropological definition of environment	anthropological definition ~~of~~ environment *naturality*	-0.15 (-26%)
right pelvic pain causes	**causes** pelvic pain **right** *ordering*	-0.18 (-46%)
define visceral	**what is** visceral *paraphrasing*	-0.26 (-38%)

In our work, we quantify the extent to which different retrieval models are susceptible to different types of query variations as measured by their drop in retrieval effectiveness. In contrast to prior works that either analyze behaviour of models when faced with modifications to the documents [31], analyze models through the lens of IR axioms [12,47] or analyze NLP models via general natural language text adversarial examples [21,48], we instantiate our *query variations* based on user-created data. Concretely, we manually label a large fraction of UQV100 queries[2] and extract six types of frequently occurring query transitions: *gen./specialization*, *aspect change*, *misspelling*, *naturality*, *ordering* and *paraphrasing*—an example of each is shown in Table 1. The last four of these categories change the query syntax but not its semantics. For each of the four syntax-changing categories, we develop automated approaches that enable us to generate query variations of each category for any input query. With these

[2] To our knowledge, UQV100 is the only publicly available dataset that contains a large number of query variations for a set of information needs.

query variation generators in place, we conduct extensive empirical work on the TREC-DL-2019 [15] and ANTIQUE [23] datasets to answer the question: *Are retrieval pipelines robust to different variations in queries that do not change its semantics?* To this end we consider seven ranking approaches: two lexical models (BM25 [49] and RM3 [1]), two neural re-ranking approaches that do not make use of transformers (KNRM [54] and cKNRM [16]) and three transformer-based re-ranking approaches (EPIC [32], BERT [40] and T5 [41]).

We find that the four types of syntax-changing query variations differ in the extent to which they degrade retrieval effectiveness: *misspellings* have the largest effect (with an average drop of 0.25 nDCG@10 points across seven retrieval models for TREC-DL-2019) while the *word ordering* has the least effect (with an average drop of nDCG@10 smaller than 0.01 for TREC-DL-2019).

Our work indicates that more research is required to improve the robustness of retrieval pipelines. Evaluation benchmarks should aim to have multiple query variations for the same information need; we provide here a number of methods to automatically generate such query variations for any dataset.

2 Related Work

Query Variation. A number of studies have argued that evaluation in IR tasks should take into account multiple instantiations of the same information need, i.e. query variations, due to their impact on the effectiveness of ranking models [4–7, 11, 36, 50, 60]. Zuccon et al. [60] proposed a mean-variance framework to explicitly take into account query variations when comparing different IR systems. Bailey et al. [6] argued that a model should be consistent to different query variations, and proposed a measure of consistency which gives additional information to effectiveness measurements.

Besides a better evaluation of models, query variations can also be employed to improve the overall effectiveness of ranking models, for instance by combining the different rankings obtained from them [8, 10] or by modelling relevance of multiple query variations [28]. They have also shown to been helpful for the problem of query performance prediction [57].

Different methods to automatically generate query variations have been proposed. Benham et al. [9] proposed to obtain query expansions through a relevance model which is built by issuing the original query against an external corpora and expanding it with additional terms from the set of external feedback documents. Lu et al. [28] employed a query-url click graph and generated query variations automatically using a two-step backward walk process. Chakraborty et al. [13] generated query variations based on an external knowledge base with a prior term distribution and by building a relevance model in an iterative manner.

Our work differs from previous work in the following ways: (I) our methods do not require access to external corpora, a relevance model or a query-url click graph; (II) we are not concerned with generating queries with the sole purpose of improving effectiveness, but in generating queries that are likely to occur in practice; and (III) each of our generator methods follows a category of our taxonomy of query variations which allows us to *diagnose* ranking models' effectiveness by analyzing what types of variations are more detrimental to what ranking models.

Model Understanding. The success of pre-trained transformer-based language models such as BERT [17] and T5 [46] on several IR benchmarks—a comprehensive account of the effectiveness gains can be found in [27]—has lead to research on understanding their behaviour and the reasons behind their significant gains in ranking effectiveness [12,31,42,45,58].

Câmara and Hauff [12] showed that BERT does not adhere to IR axioms, i.e., heuristics that a reasonable IR model should fulfill, through the use of diagnostic datasets. MacAvaney et al. [31] expanded on the axiomatic diagnostic datasets [47] with ABNIRML, a framework to understand the behaviour of neural ranking models using three different strategies: measure and match (controlling certain measurements such as term frequency and changing another), manipulation of the documents' text (e.g., by shuffling words) and through the transfer of Natural Language Processing (NLP) datasets (e.g., by comparing documents that are more/less formal). We expand on [31] by proposing textual manipulations—unlike previous methods we are inspired by *user-created* variations—to the queries instead of the documents and examine the robustness in terms of effectiveness of ranking models to such manipulations.

A different direction of research in NLP has challenged how well current evaluation schemes are actually evaluating the desired capabilities of the models through the use of held-out test sets. For example, Gardner et al. [21] proposed the manual creation of contrast sets—small perturbations that preserve artifacts but change the true label—in order to evaluate the models' decision boundaries for different NLP tasks. They showed that the model effectiveness on such contrast sets can be up to 25% lower than on the original test sets. Inspired by behavioral testing, i.e. validating input output behaviour without knowledge about internal structure, from software engineering tests, Ribeiro et al. [48] proposed to test NLP models with three different types of tests: minimum functionality tests (simple examples where the model should not fail), label (such as positive, negative and neutral in sentiment analysis) invariant changes to the input, and modifications to the input with known outcomes. With such tests at hand they were able to find actionable failures in different commercial NLP models that had already been extensively tested. It has also been shown that neural models developed for different NLP tasks can be tricked by adversarial examples [2,20,22], i.e. examples with perturbations indiscernible by humans which are misclassified by the model. In terms of query modifications, [53,59] found typos to be detrimental to the effectiveness of neural rankers. Ma et al. [29] showed that contrastive fine-tuning improves the robustness of ranking models to paraphrased and perturbed queries. Wu et al. [53] analyzed the robustness of neural rankers with respect to three dimensions: difficult queries from similar distributions, out-of-domain cases, and defense against adversarial operations. Our work differs from the adversarial line of research by evaluating the robustness of models to query modifications that could be generated by humans, i.e. transformations that naturally occur, and not modifications optimized to trick neural models.

3 Automatic Query Variations

We now first describe how we arrived at our query variation categories in a data-driven manner by annotating a large set of user-created query variations from UQV100. We end up with six categories: four that change the syntax (but not the semantics) and two that change the semantics. **In our work, we focus on the four syntax-changing categories.** We subsequently describe our methods to automatically generate the four types of syntax-changing query variations.

3.1 UQV Taxonomy

In order to better understand how queries differ when we compare different query variations for the same information need, we resort to analyzing variations from the UQV100 dataset. UQV100 contains query variations for 100 (sub)-topics from the TREC 2013 and 2014 web tracks, written by crowd workers who received a "backstory" for each topic as a starting point. On average, UQV100 contains 57.7 spelling corrected (corrected by the UQV100 authors using the spelling service of the Bing search engine) query variations per topic. We consider a query variation pair $\{q_i, q_j\}$ to be a set of two queries q_i and q_j that were provided in UQV100 for the same backstory. In total, 365K such pairs exist; Table 2 (4th column) contains a number of $\{q_i, q_j\}$ examples. We sampled 100 query variation pairs for manual annotation. Three authors of this paper (the "annotators") performed an open card sort [52]. The annotators independently sorted the query variation pairs into different piles and named them, each representing a transformation T that can be applied to q_i and then leads to q_j, i.e. $T(q_i) = q_j$. Multiple transformations might be applied to q_i in order to yield q_j, e.g. $T_2(T_1(q_i)) = q_j$.

After the independent sorting step, the different piles were discussed and merged where necessary, which yielded five categories of transformations. Since the UQV100 data used had already been spelling-corrected by its authors, we added the category *misspellings*. The resulting taxonomy can be found in Table 2. It contains a concrete definition and examples for each of our—in total—six categories: (I) *generalization or specialization*, (II) *aspect change*, (III) *misspelling*, (IV) *naturality*, (V) *word ordering* and (VI) *paraphrasing*. We observed two broad types of transformations: transformations that change the semantics of the query and transformations that do not change the semantics. The *gen./specialization* and *aspect change* transformations fall into the former type, whereas all other categories fall into the latter. We highlight here that unlike previous categorizations that describe how users revise queries in e-commerce [3,24], how to generate better queries to substitute the original query [26], how users reformulate queries in a session [25], we study here how to categorize *query variations* for the same information need which is a related but different problem.

Having arrived at our six categories, our annotators then labeled an additional set of 550 $\{q_i, q_j\}$ randomly sampled pairs from UQV100 in order to determine the distribution of these categories in UQV100. Each $\{q_i, q_j\}$ was labelled

Table 2. Taxonomy of query variations derived from a sample of the UQV100 dataset. Last column is the count of each query variation found on UQV100 based on manual annotation of tuples of queries for the same information need. Categories in grey change the semantics. * typos were already fixed for the UQV100 pairs.

Category	**Definition**	$\{q_i, q_j\}$ **from UQV100**			**Count**
Gen./specialization	Generalizes or specializes within the same information need.	american civil war	↔	number of battles in south carolina during civil war	172
Aspect change	Moves between related but different aspects within the same information need.	what types of spiders can bite you while gardening	↔	signs of spider bite	111
Misspelling	Adds or removes spelling errors.	raspberry pi	↔	raspeberry pi	*
Naturality	Moves between keyword queries and natural language queries.	how does zinc relate to wilson's disease	↔	zinc wilson's disease	118
Ordering	Changes the order of words	carotid cavernous fistula treatment.	↔	treatment carotid cavernous fistula	37
Paraphrasing	Rephrases the query by modifying one or more words.	cures for a bald spot	↔	cures for baldness	215

as belonging to one (or more) of the five categories (with the exception of *misspelling* which, as already stated, had already been corrected by the UQV100 authors). In order to determine the inter-annotator agreement, 25 $\{q_i, q_j\}$ pairs were labelled by all three annotators, and 175 pairs were each labelled by a single annotator. The inter-annotator agreement [14] was moderate (Cohen's $\kappa = 0.42$); the disagreements were highest for the *naturality* and *paraphrasing* categories. We found that a total of 56 $\{q_i, q_j\}$ pairs had more than one category assigned to it[3]. The resulting distribution is shown in Table 2 (right-most column); the categories of query variations that change the query without changing its semantics account for 57% of all the transformations. In contrast, 43% of query variations are semantic changes. Among the syntax-changing categories, we found *naturality* to be the most common with 33% of all transformations falling into this category. Having observed that query variations change the syntax, but not the semantics for the majority of cases, **we focus in the remainder of our work on syntax-changing query variations**. We leave the exploration of query variation generators for *gen./specialization* and *aspect change* as future work.

3.2 Query Generators

For each of the four syntax-changing categories, we explored different methods that generate query variations of the specified category. After an initial

[3] For example, the pair {*"what is doctor zhivago all about"*, *"dr zhivago synopsis"*} had both *paraphrasing* and *naturality* labels, as it goes from a natural language question to a keyword-base question and also paraphrases *"doctor [...] all about"* to *"dr [...] synopsis"*.

exploration of different query generator methods for each category, and filtering approaches that did not generate valid variations for the category and approaches that have high correlation with each other, we employed a total of ten different methods. These methods are listed in Table 3, each with an example transformation. We explain each one in more detail in this section. A method M_C receives as input a query q and outputs a query variation $\hat{q}$ for the category C: $M_C(q) = \hat{q}$.

Table 3. Example of applying each query generation method M for the query '*what is durable medical equipment consist of*' from TREC-DL-2019. Rightmost columns indicate the total percentage of valid queries by automatic query variation method based on manual annotation of queries from the test sets of TREC-DL-2019 and ANTIQUE.

C	Method name	M('***what is durable medical equipment consist of***')	**TREC**	**ANT**
Misspelling	`NeighbCharSwap`	*what is durable* ***medical*** *equipment consist of*	100.00%	99.50%
	`RandomCharSub`	*what is durable* ***medical*** *equipment consist of*	97.67%	91.00%
	`QWERTYCharSub`	*what is durable medical equipment* ***consist*** *of*	97.67%	98.50%
Naturality	`RmvStopWords`	~~***what is***~~ *durable medical equipment consist* ~~***of***~~	86.05%	99.50%
	`T5DescToTitle`	~~***what is***~~ *durable medical equipment* ~~**consist of**~~	81.40%	68.00%
Ordering	`RandOrderSwap`	***medical*** *is durable* ***what*** *equipment consist of*	100.00%	100.00%
Paraphrasing	`BackTransl`	*what is* ***sustainable*** *medical equipment* ~~consist of~~	53.49%	46.50%
	`T5QQP`	*what is durable medical equipment* ~~***consist of***~~	60.47%	52.50%
	`WEmbedSynSwap`	*what is durable* ***medicinal*** *equipment consist of*	62.79%	62.00%
	`WNetSynSwap`	*what is* ***long lasting*** *medical equipment consist of*	37.21%	35.50%

While most of the methods can generate multiple variations for a single input query (for example by replacing different words of the same query by synonyms or by including several spelling mistakes), for the experiments in the paper we resort to using a single query variation per method which already yields enough data for analysis (see §4). Inspired by adversarial examples, we aim to make minimal perturbations to the input text when possible, e.g. replace only one word by a synonym, thus increasing the chances of obtaining valid variations.

Misspelling. The three methods in this category add one spelling error to the query; the query term an error is introduced in is chosen uniformly at random.

`NeighbCharSwap` Swaps two neighbouring characters from a random query term (excluding stopwords[4]).

`RandomCharSub` Replaces a random character from a random query term (excluding stopwords) with a randomly chosen new ASCII character.

`QWERTYCharSub` Replaces a random character of a random query term (excluding stopwords) with another character from the QWERTY keyboard such that only characters in close proximity are chosen, replicating errors that come from typing too quickly.

Naturality. The two methods in this category transform natural language queries into keyword queries.

`RmvStopWords` Removes all stopwords from the query.

`T5DescToTitle` Applies an encoder-decoder transformer model (here we employ T5 [46]) that we fine-tuned on the task of generating the title of a TREC topic based on the TREC topic description (an example title and description tuple from `trec-robust04` is '*Evidence that rap music has a negative effect on young people.*' → '*Rap and Crime*'). We collect pairs of title and description from eleven datasets available through the IR datasets library [33].[5] Overall, we fine-tuned our model on 1322 such description/title tuples.

Ordering. In this category, we employ only one method to shuffle words as done by previous research on the order of words [31,44].

`RandOrderSwap` Randomly swap two words of the query.

Paraphrasing. The four methods in this category change one or more query terms in the process of paraphrasing.

`BackTransl` Applies a translation method to the query to a pivot language, i.e. an auxiliary language, and from the pivot language back to the original language of the query (in our case: English). In our experiments we employ the M2M100 [18] model, a multilingual model that can translate between any pair of 100 languages, and we use *'German'* as the pivot language, which yielded better results—shown by manual inspection of the generated variations—than the other two languages for which the model had the most data for training (*'Spanish'* and *'French'*). This technique has been used before as a way to generate paraphrases [19,35].

[4] We use the NLTK english stopwords list for all the methods; it is available at https://www.nltk.org/.

[5] Concretely, we made use of trec-robust04, trec-tb-2004, aquaint/trec-robust-2005, gov/trec-web-2002, ntcir-www-2, ntcir-www-3, trec-misinfo-2019, cord19/trec-covid, dd-trec-2015, dd-trec-2016 and dd-trec-2017.

`T5QQP` Applies an encoder-decoder transformer model (T5 [46]) that was fine-tuned on the task of generating a paraphrase question from the original question[6]. The model employs the Quora Question Pairs[7] dataset for fine-tuning, which has 400k pairs of questions like the following: '*How do you start a bakery?*' → '*How can one start a bakery business?*'. We also tested T5 models fine-tuned for PAWS [56] and the combination of PAWS and Quora Question Pairs, but the manual inspection of the generated queries revealed that T5 fine-tuned for Quora Question Pairs generated a higher number of valid variations.

`WEmbedSynSwap` Replaces a non-stop word by a synonym as defined by the nearest neighbour word in the embedding space according to a counter fitted-Glove embedding which yields better synonyms than standard Glove embeddings [38].

`WNetSynSwap` Replaces a non-stop word by a the first synonym found on WordNet[8]. If there are no words with valid synonyms it will not output a variation.

4 Experimental Setup

Datasets. We consider the following datasets: TREC-DL-2019 [15] for the passage retrieval task and ANTIQUE [23] for the non-factoid question answering task. They have 367,013/5,193/43 and 2,426/-/200 instances respectively for training, validation and test. The queries from TREC-DL-2019 are smaller on average: 5.51 terms vs 10.51 from ANTIQUE. For each of the test set queries, we generate one query variation by each generator method, and we use only the valid query variations in our experiments (according to manual annotation), leading to 334 and 1,706 valid query variations for TREC-DL-2019 and ANTIQUE.

Ranking Models. We use different ranking models that range from traditional lexical models, such as BM25, to neural ranking models, such as KNRM and neural ranking models that employ transformer-based language models, such as BERT. For all of our experiments, we apply BM25 as a first stage retriever and re-rank the top 100 results with the neural ranking models, which is an established and efficient approach [27].

For **BM25** [49] and **RM3** [1] we resort to the default hyperparameters and implementation provided by the PyTerrier toolkit [34]. We trained the kernel-based ranking models **KNRM** [54] and **cKNRM** [16] on the training sets of TREC-DL-2019 and ANTIQUE using default settings from the OpenNIR [30] implementation. For the BERT-based methods **EPIC** [32], an efficiency focused model that encodes query and documents separately, and **BERT** [40], also known as monoBERT, which concatenates query and document and makes predictions based on the `[CLS]` token representation, we fine-tune the `bert-base-uncased` model for the train datasets. For **T5** [46] we use

[6] As available here https://huggingface.co/ramsrigouthamg/t5_paraphraser.

[7] https://www.kaggle.com/c/quora-question-pairs.

[8] https://wordnet.princeton.edu/.

the monoT5 [41] implementation of the PyTerrier T5 plugin[9] which has the pre-trained weights for MSMarco [39] by the authors of monoT5.

Query Generators Implementation. As for our methods of generating query variations, for `T5DescToTitle` and `T5QQP` we rely on pre-trained T5 models (`t5-base`) and we fine-tune them using the Huggingface transformers library [51]. For `BackTransl` we use the `facebook/m2m100_418M` pre-trained model from the transformers library[10]. For all other methods, we use the implementations from the TextAttack [37] library.

Quality of Query Generators. Given the automatic nature of the methods we introduced, we need to evaluate their quality. To this end, we consider two properties of the generated queries: (I) $\hat{q}$ maintains the same semantics as q, and (II) the syntax difference between q and $\hat{q}$ can be attributed to the category. All pairs of q and $\hat{q} = M(q)$ from the test sets of TREC-DL-2019 (43 queries) and ANTIQUE (200 queries) for each of the 10 automatic variation methods went through the following process. First, we automatically set the variations from *misspelling*[11] and *ordering* as valid, since they are rule based transformations to the input. Then all transformations that generate a variation that is identical to the input query ($\hat{q} = M(q) = q$) was automatically set to invalid. Three authors then annotated independently the remaining 1,371 pairs of $\{q, \hat{q}\}$ for the two mentioned properties (binary labels). The percentage of queries that are valid (i.e. they have both desired properties) are displayed in the right-most columns of Table 3 for the 10 automatic variation methods used in the paper and all 2,430 combinations of $\{q, \hat{q}\}$. We find the methods in the *paraphrasing* category to yield the largest percentage of invalid query variations: fewer than 38% of query variations generated via `WNetSynSwap` are valid. A manual inspection of the invalid queries reveal the following insights: (I) `T5DescToTitle` at times removes query terms that are important for the query and thus change its semantics (e.g. '*if i had a bad breath what should i do*' → '*if i had a*'), (II) `BackTransl` and `T5QQP` methods can generate an identical copy of the input query which was automatically labelled as invalid and (III) transformations that replace words by their presumed synonyms (`WEmbedSynSwap` and `WNetSynSwap`) at times adds words that are not in fact synonymous in the query context (e.g. '*what is dark energy*' → '*what is blackness energy*' and '*what is a active margin*' → '*what is a active border*').

To evaluate the robustness of the ranking models, we resort to using only the valid queries as defined by the manual annotations. Overall, we have thus 2,040 valid queries for datasets TREC-DL-2019 and ANTIQUE that we employ in the experiments that follow.

[9] https://github.com/terrierteam/pyterrier_t5.

[10] https://huggingface.co/facebook/m2m100_418M.

[11] *misspelling* methods can generate invalid queries when all words of the query are stop-words (e.g. '*how is it being you*' from ANTIQUE would generate the same query as output since there is no non stop-words to modify).

5 Results

To explore the robustness of our three types of ranking models we compare the effectiveness of our models when we replace the original query with the respective query variation. The results of this experiment are displayed in Table 4 for both the TREC-DL-2019 and ANTIQUE datasets. Each row shows the effectiveness of the ranking models (columns) when using the queries obtained from each automatic query variation method. The last column ($\#Q$) displays the number of valid queries generated by each query variation method; the invalid queries are replaced with the original ones[12].

The results show that for most of the query variations and ranker combinations we observe a statistical significant effectiveness drop (49 out of 70 times for TREC-DL-2019 and 54 out of 70 times for ANTIQUE), and that no set of query variations improves statistically over using the original query. If we look into the percentage of overall effectiveness decreases considering only the valid queries, we see on average that the models become 20.62% and 19.21% less effective for TREC-DL-2019 and ANTIQUE respectively. **This answers our research question indicating that retrieval pipelines are not robust to query variations**. This confirms previous empirical evidence that query variations induce a big variability effect on different IR systems [6,60]. We show that even with newer large-scale collections such as TREC-DL-2019, retrieval pipelines are not robust to such variations.

There are several potential explanations for this drop in effectiveness besides the lack of robustness of neural rankers. The first-stage ranker may be the point of failure, being unable to retrieve sufficiently many relevant documents for the neural rankers to re-rank. It is also possible that the query variations lead to unjudged documents being ranked highly by the retrieval pipelines, which in the standard retrieval evaluation setup are considered non-relevant. We now present two experiments to show that these alternative explanations are not the cause in drop of retrieval effectiveness.

Let's focus first on the first-stage ranker. We first calculated the average drop in effectiveness when we increase the re-ranking threshold. While the number of documents in the re-ranking set increases[13], neural models still struggle, e.g. for BERT the nDCG@10 decreases on average by 40%, 34% and 31%[14]. This indicates that even if we increase the number of relevant documents to be re-ranked, neural rankers still fail when faced with query variations.

To further isolate the effect of the first-stage retrieval module, we analyzed whether the effectiveness of the pipelines would not degrade in case the first-stage retrieval was performed on the original query. In this experiment only the re-ranker models use the query variations and we check whether the effectiveness drops persist. The results reveal that there are still statistically significant

[12] While rows are directly comparable, methods with fewer valid queries are a lower bound of the potential decreases in effectiveness.

[13] BM25 has R@10, R@100 and R@1000 of 0.06, 0.25 and 0.48 for *misspelling*.

[14] Similar results are obtained for other neural rankers.

Table 4. Effectiveness (nDCG@10) of different methods for TREC-DL-2019 and ANTIQUE when faced with different query variations. Bold indicates the highest values observed for each model and ↓/↑ subscripts indicate statistically significant losses/improvements, using two-sided paired Student's T-Test at 95% confidence interval with Bonferroni correction when compared against the model with original queries. #*Q* is the number of valid query variations (invalid query variations are replaced by the original query).

					TREC-DL-2019			
Category	**Variation**	**BM25**	**RM3**	**KNRM**	**cKNRM**	**EPIC**	**BERT**	**T5**	**#Q**
–	original query	**0.480**	**0.516**	**0.502**	**0.493**	**0.624**	**0.645**	0.700	43
Misspelling	NeighbCharSwap	0.275$^{\downarrow}$	0.275$^{\downarrow}$	0.316$^{\downarrow}$	0.309$^{\downarrow}$	0.389$^{\downarrow}$	0.416$^{\downarrow}$	0.495$^{\downarrow}$	43
	RandomCharSub	0.231$^{\downarrow}$	0.233$^{\downarrow}$	0.236$^{\downarrow}$	0.226$^{\downarrow}$	0.295$^{\downarrow}$	0.328$^{\downarrow}$	0.396$^{\downarrow}$	42
	QWERTYCharSub	0.244$^{\downarrow}$	0.250$^{\downarrow}$	0.267$^{\downarrow}$	0.297$^{\downarrow}$	0.351$^{\downarrow}$	0.387$^{\downarrow}$	0.446$^{\downarrow}$	42
Naturality	RmvStopWords	0.478	0.511	0.484	0.476	0.621	0.639	0.687	37
	T5DescToTitle	0.421	0.434$^{\downarrow}$	0.392	0.393	0.506$^{\downarrow}$	0.536$^{\downarrow}$	0.571$^{\downarrow}$	35
Ordering	RandOrderSwap	0.480	0.516	0.502	0.471	0.623	0.635	0.697	43
Paraphrasing	BackTransl	0.396	0.420$^{\downarrow}$	0.393	0.361$^{\downarrow}$	0.530	0.547$^{\downarrow}$	0.606	23
	T5QQP	0.472	0.504	0.454	0.461	0.605	0.640	**0.705**	26
	WEmbedSynSwap	0.353$^{\downarrow}$	0.354$^{\downarrow}$	0.382$^{\downarrow}$	0.368$^{\downarrow}$	0.475$^{\downarrow}$	0.472$^{\downarrow}$	0.560$^{\downarrow}$	27
	WNetSynSwap	0.349$^{\downarrow}$	0.365$^{\downarrow}$	0.381$^{\downarrow}$	0.361$^{\downarrow}$	0.449$^{\downarrow}$	0.447$^{\downarrow}$	0.545$^{\downarrow}$	16
					ANTIQUE				
Category	**Variation**	**BM25**	**RM3**	**KNRM**	**cKNRM**	**EPIC**	**BERT**	**T5**	**#Q**
–	original query	**0.229**	**0.217**	0.218	0.207	0.266	**0.421**	**0.334**	200
Misspelling	NeighbCharSwap	0.156$^{\downarrow}$	0.148$^{\downarrow}$	0.159$^{\downarrow}$	0.145$^{\downarrow}$	0.184$^{\downarrow}$	0.287$^{\downarrow}$	0.251$^{\downarrow}$	199
	RandomCharSub	0.162$^{\downarrow}$	0.159$^{\downarrow}$	0.156$^{\downarrow}$	0.148$^{\downarrow}$	0.189$^{\downarrow}$	0.280$^{\downarrow}$	0.249$^{\downarrow}$	182
	QWERTYCharSub	0.161$^{\downarrow}$	0.153$^{\downarrow}$	0.160$^{\downarrow}$	0.155$^{\downarrow}$	0.192$^{\downarrow}$	0.299$^{\downarrow}$	0.266$^{\downarrow}$	197
Naturality	RmvStopWords	0.227	0.216	**0.222**	**0.215**	**0.269**	0.383$^{\downarrow}$	0.320	199
	T5DescToTitle	0.167$^{\downarrow}$	0.165$^{\downarrow}$	0.160$^{\downarrow}$	0.167$^{\downarrow}$	0.200$^{\downarrow}$	0.270$^{\downarrow}$	0.240$^{\downarrow}$	136
Ordering	RandOrderSwap	0.229	0.217	0.218	0.198	0.267	0.413$^{\downarrow}$	0.325$^{\downarrow}$	200
Paraphrasing	BackTransl	0.162$^{\downarrow}$	0.155$^{\downarrow}$	0.160$^{\downarrow}$	0.144$^{\downarrow}$	0.204$^{\downarrow}$	0.305$^{\downarrow}$	0.258$^{\downarrow}$	93
	T5QQP	0.220	0.207	0.210	0.196	0.261	0.393$^{\downarrow}$	0.321	105
	WEmbedSynSwap	0.176$^{\downarrow}$	0.172$^{\downarrow}$	0.190$^{\downarrow}$	0.169$^{\downarrow}$	0.214$^{\downarrow}$	0.325$^{\downarrow}$	0.283$^{\downarrow}$	124
	WNetSynSwap	0.179$^{\downarrow}$	0.175$^{\downarrow}$	0.196$^{\downarrow}$	0.177$^{\downarrow}$	0.212$^{\downarrow}$	0.324$^{\downarrow}$	0.273$^{\downarrow}$	71

effectiveness drops when only the re-ranker models use the query variations, although in smaller magnitude. While the drops in effectiveness of the pipelines when using query variations for the entire pipeline are on average of ≈ 20% in nDCG@10, when using the query variations only for re-ranking they are ≈ 9%. **This indicates that not only the first stage retrieval module is not robust to query variations, but also the neural re-rankers**.

Let's now focus on the matter of unjudged documents. It is possible that we are underestimating the effectiveness of the retrieval pipelines when facing query variations if (I) the number of unjudged documents in the top-10 ranked lists increases and (II) they turn out to be relevant. When counting the amount of judged documents in the top-10 ranked lists of the retrieval pipelines, we find that on average the number actually increases (4.30% for TREC-DL-2019 and 0.36% for ANTIQUE), **meaning that the performance drops of the**

retrieval pipelines cannot be attributed to unjudged documents being brought up in the ranking by the query variations.

6 Conclusions

We first described a taxonomy of transformations between two queries for the same information need that characterizes how exactly a query is modified to arrive at one of its variants. We found six different types of transformations, and focused on the ones that do not change the query semantics: *misspelling*, *naturality*, *ordering* and *paraphrasing*. They account for 57% of observed variations in the UQV100 dataset. For each category, we proposed different methods to automatically generate query variations. We studied the quality of the generated query variations, and analyzed how robust retrieval pipelines are to them. Our results on two datasets quantify how much each model is affected by each type of query variation, demonstrating large effectiveness drops of 20% on average when compared to the original queries. As future work, we believe that it is important to study (I) how to automatically generate valid query variation generators for categories that do change the semantics of the query and (II) techniques to improve the robustness of existing ranking pipelines.

Acknowledgements. This research has been supported by NWO projects SearchX (639.022.722) and NWO Aspasia (015.013.027).

References

1. Abdul-Jaleel, N., et al.: UMass at TREC 2004: novelty and hard. Computer Science Department Faculty Publication Series, p. 189 (2004)
2. Alzantot, M., Sharma, Y., Elgohary, A., Ho, B.J., Srivastava, M., Chang, K.W.: Generating natural language adversarial examples. arXiv preprint arXiv:1804.07998 (2018)
3. Amemiya, Y., Manabe, T., Fujita, S., Sakai, T.: How do users revise zero-hit product search queries? In: Hiemstra, D., Moens, M.-F., Mothe, J., Perego, R., Potthast, M., Sebastiani, F. (eds.) ECIR 2021. LNCS, vol. 12657, pp. 185–192. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-72240-1_14
4. Bailey, P., Moffat, A., Scholer, F., Thomas, P.: User variability and IR system evaluation. In: Proceedings of The 38th International ACM SIGIR conference on research and development in Information Retrieval, pp. 625–634 (2015)
5. Bailey, P., Moffat, A., Scholer, F., Thomas, P.: Uqv100: a test collection with query variability. In: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, pp. 725–728 (2016)
6. Bailey, P., Moffat, A., Scholer, F., Thomas, P.: Retrieval consistency in the presence of query variations. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 395–404 (2017)
7. Belkin, N.J., Cool, C., Croft, W.B., Callan, J.P.: The effect multiple query representations on information retrieval system performance. In: Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 339–346 (1993)

8. Belkin, N.J., Kantor, P., Fox, E.A., Shaw, J.A.: Combining the evidence of multiple query representations for information retrieval. Inf. Process. Manage. **31**(3), 431–448 (1995)
9. Benham, R., Culpepper, J.S., Gallagher, L., Lu, X., Mackenzie, J.: Towards efficient and effective query variant generation. In: DESIRES, pp. 62–67 (2018)
10. Benham, R., Mackenzie, J., Moffat, A., Culpepper, J.S.: Boosting search performance using query variations. ACM Trans. Inf. Syst. (TOIS) **37**(4), 1–25 (2019)
11. Buckley, C., Walz, J.: The TREC-8 query track. In: TREC (1999)
12. Câmara, A., Hauff, C.: Diagnosing BERT with retrieval heuristics. In: Jose, J.M., et al. (eds.) ECIR 2020. LNCS, vol. 12035, pp. 605–618. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-45439-5_40
13. Chakraborty, A., Ganguly, D., Conlan, O.: Retrievability based document selection for relevance feedback with automatically generated query variants. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp. 125–134 (2020)
14. Cohen, J.: A coefficient of agreement for nominal scales. Educ. Psychol. Measur. **20**(1), 37–46 (1960)
15. 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)
16. Dai, Z., Xiong, C., Callan, J., Liu, Z.: Convolutional neural networks for soft-matching N-grams in ad-hoc search. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, pp. 126–134 (2018)
17. Devlin, J., Chang, M.W., 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, Volume 1 (Long and Short Papers), pp. 4171–4186 (2019)
18. Fan, A., et al.: Beyond English-centric multilingual machine translation. arXiv preprint arXiv:2010.11125 (2020)
19. Federmann, C., Elachqar, O., Quirk, C.: Multilingual whispers: generating paraphrases with translation. In: Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019), pp. 17–26 (2019)
20. Gao, J., Lanchantin, J., Soffa, M.L., Qi, Y.: Black-box generation of adversarial text sequences to evade deep learning classifiers. In: 2018 IEEE Security and Privacy Workshops (SPW), pp. 50–56. IEEE (2018)
21. Gardner, M., et al.: Evaluating models' local decision boundaries via contrast sets. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, pp. 1307–1323 (2020)
22. Garg, S., Ramakrishnan, G.: Bae: Bert-based adversarial examples for text classification. arXiv preprint arXiv:2004.01970 (2020)
23. Hashemi, H., Aliannejadi, M., Zamani, H., Croft, W.B.: ANTIQUE: a non-factoid question answering benchmark. In: Jose, J.M., et al. (eds.) ECIR 2020. LNCS, vol. 12036, pp. 166–173. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-45442-5_21
24. Hirsch, S., Guy, I., Nus, A., Dagan, A., Kurland, O.: Query reformulation in e-commerce search. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1319–1328 (2020)
25. Jansen, B.J., Booth, D.L., Spink, A.: Patterns of query reformulation during web searching. J. Am. Soc. Inf. Sci. Technol. **60**(7), 1358–1371 (2009)

26. Jones, R., Rey, B., Madani, O., Greiner, W.: Generating query substitutions. In: Proceedings of the 15th international conference on World Wide Web, pp. 387–396 (2006)
27. Lin, J., Nogueira, R., Yates, A.: Pretrained transformers for text ranking: Bert and beyond. arXiv preprint arXiv:2010.06467 (2020)
28. Lu, X., Kurland, O., Culpepper, J.S., Craswell, N., Rom, O.: Relevance modeling with multiple query variations. In: Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval, pp. 27–34 (2019)
29. Ma, X., Santos, C.N.d., Arnold, A.O.: Contrastive fine-tuning improves robustness for neural rankers. arXiv preprint arXiv:2105.12932 (2021)
30. MacAvaney, S.: OpenNIR: a complete neural ad-hoc ranking pipeline. In: WSDM 2020 (2020)
31. MacAvaney, S., Feldman, S., Goharian, N., Downey, D., Cohan, A.: Abnirml: analyzing the behavior of neural IR models. arXiv preprint arXiv:2011.00696 (2020)
32. MacAvaney, S., Nardini, F.M., Perego, R., Tonellotto, N., Goharian, N., Frieder, O.: Expansion via prediction of importance with contextualization. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1573–1576 (2020)
33. MacAvaney, S., Yates, A., Feldman, S., Downey, D., Cohan, A., Goharian, N.: Simplified data wrangling with ir_datasets. In: SIGIR (2021)
34. Macdonald, C., Tonellotto, N.: Declarative experimentation in information retrieval using pyterrier. In: Proceedings of ICTIR 2020 (2020)
35. Mallinson, J., Sennrich, R., Lapata, M.: Paraphrasing revisited with neural machine translation. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pp. 881–893 (2017)
36. Moffat, A., Scholer, F., Thomas, P., Bailey, P.: Pooled evaluation over query variations: users are as diverse as systems. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp. 1759–1762 (2015)
37. Morris, J., Lifland, E., Yoo, J.Y., Grigsby, J., Jin, D., Qi, Y.: 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, pp. 119–126 (2020)
38. Mrkšić, N., et al.: Counter-fitting word vectors to linguistic constraints. arXiv preprint arXiv:1603.00892 (2016)
39. Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., Deng, L.: Ms marco: a human generated machine reading comprehension dataset. In: CoCo@ NIPS (2016)
40. Nogueira, R., Cho, K.: Passage re-ranking with bert. arXiv preprint arXiv:1901.04085 (2019)
41. Nogueira, R., Jiang, Z., Lin, J.: Document ranking with a pretrained sequence-to-sequence model. arXiv preprint arXiv:2003.06713 (2020)
42. Padigela, H., Zamani, H., Croft, W.B.: Investigating the successes and failures of Bert for passage re-ranking. arXiv preprint arXiv:1905.01758 (2019)
43. Penha, G., Hauff, C.: Curriculum learning strategies for IR. In: Jose, J.M., Yilmaz, E., Magalhães, J., Castells, P., Ferro, N., Silva, M.J., Martins, F. (eds.) ECIR 2020. LNCS, vol. 12035, pp. 699–713. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-45439-5_46

44. Pham, T.M., Bui, T., Mai, L., Nguyen, A.: Out of order: How important is the sequential order of words in a sentence in natural language understanding tasks? arXiv preprint arXiv:2012.15180 (2020)
45. Qiao, Y., Xiong, C., Liu, Z., Liu, Z.: Understanding the behaviors of Bert in ranking. arXiv preprint arXiv:1904.07531 (2019)
46. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv preprint arXiv:1910.10683 (2019)
47. Rennings, D., Moraes, F., Hauff, C.: An axiomatic approach to diagnosing neural IR models. In: Azzopardi, L., Stein, B., Fuhr, N., Mayr, P., Hauff, C., Hiemstra, D. (eds.) ECIR 2019. LNCS, vol. 11437, pp. 489–503. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-15712-8_32
48. Ribeiro, M.T., Wu, T., Guestrin, C., Singh, S.: Beyond accuracy: behavioral testing of NLP models with checklist. arXiv preprint arXiv:2005.04118 (2020)
49. Robertson, S.E., Walker, S.: Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In: SIGIR 1994, pp. 232–241. Springer, Cham (1994). https://doi.org/10.1007/978-1-4471-2099-5_24
50. Spark-Jones, K.: Report on the need for and provision of an 'ideal' information retrieval test collection. Computer Laboratory (1975)
51. Wolf, T., et al.: Transformers: state-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 38–45. Association for Computational Linguistics, Online, October 2020. https://www.aclweb.org/anthology/2020.emnlp-demos.6
52. Wood, J.R., Wood, L.E.: Card sorting: current practices and beyond. J. Usabil. Stud. **4**(1), 1–6 (2008)
53. Wu, C., Zhang, R., Guo, J., Fan, Y., Cheng, X.: Are neural ranking models robust? arXiv preprint arXiv:2108.05018 (2021)
54. Xiong, C., Dai, Z., Callan, J., Liu, Z., Power, R.: End-to-end neural ad-hoc ranking with kernel pooling. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 55–64 (2017)
55. Yang, W., Zhang, H., Lin, J.: Simple applications of Bert for ad hoc document retrieval. arXiv preprint arXiv:1903.10972 (2019)
56. Yang, Y., Zhang, Y., Tar, C., Baldridge, J.: PAWS-X: a cross-lingual adversarial dataset for paraphrase identification. In: Proceedings of EMNLP (2019)
57. Zendel, O., Shtok, A., Raiber, F., Kurland, O., Culpepper, J.S.: Information needs, queries, and query performance prediction. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 395–404 (2019)
58. Zhan, J., Mao, J., Liu, Y., Zhang, M., Ma, S.: An analysis of Bert in document ranking. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1941–1944 (2020)
59. Zhuang, S., Zuccon, G.: Dealing with typos for Bert-based passage retrieval and ranking. arXiv preprint arXiv:2108.12139 (2021)
60. Zuccon, G., Palotti, J., Hanbury, A.: Query variations and their effect on comparing information retrieval systems. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, pp. 691–700 (2016)