Evaluating the Robustness of Retrieval Pipelines with Query Variation Generators

Gustavo Penha  
g.penha-1@tudelft.nl  
Delft University of Technology  
Delft, Netherlands

Arthur Câmara  
a.barbosacamara@tudelft.nl  
Delft University of Technology  
Delft, Netherlands

Claudia Hauff  
c.hauff@tudelft.nl  
Delft University of Technology  
Delft, Netherlands

ABSTRACT

Heavily pre-trained transformers for language modelling, such as BERT, have shown to be remarkably effective for Information Retrieval (IR) tasks. IR benchmarks evaluate the effectiveness of (neural) ranking models 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 easily make mistakes when tested with adversarial examples, i.e., examples with minimal modifications that do not change its label. Motivated by those observations we aim to answer the following question with our work: 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 (specifically UQV100) of user created query variations. For example, from the query ‘cures for a bald spot’ to the variation ‘cures for baldness’ we are applying a paraphrasing transformation that replaces words with synonyms. For each syntax-changing category of our taxonomy, we employ different automatic methods that when applied to a query generate a query variation. We conduct experiments on two datasets (TREC-DL-2019 and ANTIQUE) and create a total of 2430 query variations from 243 topics across both datasets. Our experimental results for two different IR tasks reveal that retrieval pipelines are not robust to query variations that maintain the content the same, with effectiveness drops of ~20% on average when compared with the original query as provided in the datasets. Our findings indicate that further work is required to make retrieval pipelines with neural ranking models more robust and that IR collections should include query variations, e.g., using the methods proposed here, for a single information need to better understand models capabilities. 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 [19] have been shown to be remarkably effective for a wide range of Information Retrieval (IR) tasks [41, 44, 58]. 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. However, previous research has shown that queries created by users given a fixed information need may vary widely [6, 63]. In the UQV100 [5] dataset for instance, crowd workers on average created 57.7 unique queries for a given information need as instantiated as a backstory, e.g., “You have heard quite a lot about cheap computing as being the way of the future, including one recent model called a Raspberry Pi. You start thinking about buying one, and wonder how much they cost.”

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 | -1.00 (-100%) |
| cost of interior concrete flooring | concrete flooring finishing aspect change | -0.15 (-26%) |
| what is theraderm used for | what is thrraderm used for misspelling | -0.11 (-23%) |
| anthropological definition of environment | anthropological definition of environment naturality | -0.08 (-13%) |
| right pelvic pain causes | causes pelvic pain right ordering | -0.18 (-46%) |
| define visceral | what is visceral paraphrasing | -0.26 (-38%) |

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 anecdotaly shown in Table 1 for a vanilla BERT model for ranking [41]. 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%.

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 [32], analyze models through the
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, 37, 52, 63]. Zuccon et al. [63] 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 [30]. They have also shown to be helpful for the problem of query performance prediction [60].

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. [30] 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 automatically based on an external knowledge base with a prior term distribution or by building a relevance model in an iterative manner.

Our work differs from previous work on automatic query variation generation 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.

2.2 Model Understanding

The success of pre-trained transformer-based language models such as BERT [19] and T5 [47] on several IR benchmarks—a comprehensive account of the effectiveness gains can be found in [29]—has lead to research on understanding their behaviour and the reasons behind their significant gains in ranking effectiveness [12, 32, 43, 46, 61].

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. [32] expanded on the axiomatic diagnostic datasets [48] with ABNIRML, a framework to understand the behaviour of neural ranking models using three different strategies: measure and match (controlling certain measurements such as relevance or term frequency and changing another), manipulation of the documents’ text (for example by shuffling words or replacing it with the query) and through the transfer of Natural Language Processing (NLP) datasets (for example comparing documents that are more/less fluent or formal with inferred queries). We expand on MacAvaney et al. [32]’s work 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 neural ranking models to such manipulations.

A different direction of research in NLP has challenged how well current evaluation schemes through the use of held-out test sets are actually evaluating the desired capabilities of the models. For example, Gardner et al. [23] 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. [49] 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, 22, 24], i.e. examples with perturbations indiscernible by humans which get misclassified by the model. In terms of queries modifications, [56, 62] found typos to be detrimental to the effectiveness of neural rankers. Wu et al. [56] analyzed the robustness of neural rankers with respect to three dimensions: difficult queries from similar distribution, 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 in Section 3.1 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. In Section 3.2 we subsequently describe our methods to automatically generate query variations for each category of the taxonomy that does not change the query semantics.

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 from the 365K available ones for manual annotation. Three authors of this paper (the “annotators”) performed an open card sort [55]. 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. \( T_1(T_2(q_i)) = q_j \).

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, 26], how to generate better queries to substitute the original query [28], how users restructure queries in a session [27], 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 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\(^2\). 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.

\(^2\)For example, the pair \( \{\text{what is doctor zhivago all about} \} \) 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”
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. * spelling errors were already fixed for the UQV100 pairs.

| Category                  | Definition                                                                 | Changes Semantics | \{q_i, q_j\} Examples 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 (26.34%) |
| 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 (17.00%) |
| Misspelling               | Adds or removes spelling errors.                                            |                   | raspberry pi ↔ raspberry pi *     |           |
| Naturality                | Moves between keyword queries and natural language queries.                |                   | how does zinc relate to wilson’s disease ↔ zinc wilson’s disease         | 118 (18.07%) |
| Ordering                  | Changes the order of words                                                  |                   | carotid cavernous fistula treatment ↔ treatment carotid cavernous fistula | 37 (5.67%) |
| Paraphrasing              | Rephrases the query by modifying one or more words.                         |                   | cures for a bald spot ↔ cures for baldness                            | 215 (32.92%) |

Table 3: Example of applying each method \(M\) for the query ‘what is durable medical equipment consist of’ from TREC-DL-2019 resulting in valid examples. Rightmost columns indicate the total percentage of valid queries by automatic query variation method based on manual annotation of queries from the test sets.

| Category      | Method Name    | \(M(\text{‘what is durable medical equipment consist of’})\) | TREC-DL-2019 | ANTIQUE |
|---------------|----------------|-------------------------------------------------------------|--------------|---------|
| 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%  |
|               | QMERTYCharSub  | what is durable medical equipment consist of                 | 97.67%       | 98.50%  |
| Naturality    | RemoveStopWords| what is durable medical equipment consist of                 | 86.05%       | 99.50%  |
|               | T5DescToTitle  | what is durable medical equipment consist of                 | 81.40%       | 68.00%  |
| Ordering      | RandomOrderSwap| medical is durable what equipment consist of                | 100.00%      | 100.00% |
| Paraphrasing  | BackTranslation| what is sustainable medical equipment consist of             | 53.49%       | 46.50%  |
|               | T5QQP          | what is durable medical equipment consist of                 | 60.47%       | 52.50%  |
|               | WordEmbedSynSwap| what is durable medicinal equipment consist of               | 62.79%       | 62.00%  |
|               | WordNetSynSwap | what is long lasting medical equipment consist of            | 37.21%       | 35.50%  |

After an initial 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 \(\hat{q}\): \(M_C(q) = \hat{q}\).

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 §4.1). 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, increasing the chances of obtaining valid variations.

3.2.1 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). **RandomCharSub** Replaces a random character from a random query term (excluding stopwords) with a randomly chosen new ASCII character.
WordNetSynSwap replaces a non-stop word by the first synonym found on WordNet. If there are no words with valid synonyms it will not output a valid variation.

WordNetSynSwap Replaces a non-stop word by the first synonym

RemoveStopWords Removes all stopwords from the query.

4 EXPERIMENTAL SETUP

In this section we describe our experimental setup aimed to answer the following research question: are retrieval pipelines robust to different variations in queries that do not change its semantics?

4.1 Datasets

We consider the following datasets in our experiments: TREC-DL-2019 [16] for the passage retrieval task and ANTIQUE [25] for non-factoid question answering task, containing 43 and 200 queries respectively in their test sets. For each of the test set queries, we generate one query variation for each of the proposed methods, and we use the manual annotation described in this section (§4.4) to take into account only the valid generated query variations in our experiments. The statistics of the datasets can be found in Table 4.

4.2 Ranking Models

We use different ranking models that cover from lexical traditional models (Trad) such as BM25, to neural ranking models (NN) such as KNRM and neural ranking models that employ transformer-based language models (TNN) 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 [29].

For BM25 [50] and RM3 [1] we resort to the default hyperparameters and implementation provided by the PyTerrier toolkit [35]. We trained the kernel-based ranking models KNRM [57] and CK-NRM [18] on the training sets of TREC-DL-2019 and ANTIQUE using default settings from the OpenNIR [31] implementation. For the BERT-based methods EPIC [33], an efficiency focused model that encodes query and documents separately, and BERT [41], also known as monoBERT, which concatenates query and the document and makes predictions based on the [CLS] token representation, we fine-tune the bert-base-uncased model for the train datasets. For T5 [47] we use the monoT5 [42] implementation from PyTerrier T5 plugin which has the pre-trained weights for MSMarco [40] by the original authors of monoT5.

4.3 Query Generators Implementation

As for our methods of generating query variations, for T5DescToTitle and T5QP we rely on pre-trained T5 models (t5-base) and we fine-tune them using the Huggingface transformers library [54]. For BackTranslation we use the facebook/m2m100_418M pre-trained

Table 4: Statistics of the datasets.

| Datasets       | TREC-DL-2019 | ANTIQUE |
|----------------|--------------|---------|
| #Q train       | 367013       | 2426    |
| #Q valid       | 5193         |         |
| #Q test        | 43           | 200     |
| # terms/ Q test| 5.51         | 10.51   |
| # valid query variations | 334         | 1706    |

4.4 Query Generators Implementation

As for our methods of generating query variations, for T5DescToTitle and T5QP we rely on pre-trained T5 models (t5-base) and we fine-tune them using the Huggingface transformers library [54]. For BackTranslation we use the facebook/m2m100_418M pre-trained

4.4 Query Generators Implementation

As for our methods of generating query variations, for T5DescToTitle and T5QP we rely on pre-trained T5 models (t5-base) and we fine-tune them using the Huggingface transformers library [54]. For BackTranslation we use the facebook/m2m100_418M pre-trained
model from the transformers library\textsuperscript{8}. For all other methods, we use the implementations from the TextAttack [38] library.

### 4.4 Quality of Query Generators

Given the automatic nature of the methods we introduced, we need to evaluate their quality: how good are these methods at generating query variations that a user would also generate?

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 \( C \). 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 to the following process. First, we automatically set the variations from misspelling\textsuperscript{9} 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. The annotators (the authors) then annotated independently the remaining 1371 pairs of \( \{q, \hat{q}\} \) for the two mentioned properties (binary labels). The percentage of queries that are valid (both desired properties) are displayed at the right-most columns of Table 3 for the 10 automatic variation methods used in the paper and all combinations of \( \{q, \hat{q}\} \) (2430).

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 WordNetSynSwap are valid. A manual inspection of the invalid queries reveal the following insights: (i) T5DescToTitle\textsuperscript{10} 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) BackTranslation and T5QQP methods can generate an identical copy of the input query which was automatically labelled as invalid (e.g. ‘what is dark energy’ → ‘what is dark energy’) and (iii) transformations that replace words by their presumed synonyms (WordEmbedSynSwap and WordNetSynSwap) 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.

### 5 RESULTS

In this section we first describe our main results on the robustness of models to query variations, analyzing them by category of variation and by category of ranking model. We then move on to discussing the fusion of the ranking list obtained by the query variations.

---

\textsuperscript{8}https://huggingface.co/facebook/m2m100_418M

\textsuperscript{9}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)

\textsuperscript{10}While rows are directly comparable, methods with fewer valid queries are a lower bound of the potential decreases in effectiveness.

---

Figure 1: Distribution of nDCG@10 ∆ for different re-ranking thresholds when using BERT as a re-ranker.

5.1 Robustness to Query Variations

In order to explore the robustness of our three types of ranking models (traditional, neural and transformer-based), 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 5 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\textsuperscript{10}.

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 main 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, 63]. We show that even with newer large-scale collections such as TREC-DL-2019, pipelines with neural ranking models 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. Figure 1 shows the effect of increasing the re-ranking threshold on the distribution of nDCG@10 ∆ when using BERT, revealing that although the number...
Table 5: 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 when compared against the same model with the original queries. #Q indicates the number of valid query variations for the method (invalid query variations are replaced by the original query).

| Category       | Query Variation       | BM25   | RM3   | KNRM  | CKNRM | EPIC  | BERT  | T5    | #Q  |
|----------------|-----------------------|--------|-------|-------|-------|-------|-------|-------|-----|
| TREC-DL-2019  | original query        | 0.4795 | 0.5156| 0.5015| 0.4931| 0.6240| 0.6449| 0.6997| 43  |
|                | NeighCharSwap         | 0.2747 | 0.2748| 0.3164| 0.3085| 0.3804| 0.4161| 0.4950| 43  |
|                | RandomCharSub         | 0.2314 | 0.2333| 0.2363| 0.2263| 0.2950| 0.3283| 0.3963| 42  |
|                | QWERTYCharSub         | 0.2435 | 0.2504| 0.2672| 0.2965| 0.3512| 0.3867| 0.4458| 42  |
|                | RemoveStopWords       | 0.4778 | 0.5113| 0.4842| 0.4756| 0.6214| 0.6390| 0.6866| 37  |
|                | T5DescToTitle         | 0.4215 | 0.4344| 0.3920| 0.3928| 0.5063| 0.5361| 0.5710| 35  |
|                | BackTranslation       | 0.3964 | 0.4195| 0.3927| 0.3605| 0.5301| 0.5467| 0.6058| 23  |
|                | T5QQP                 | 0.4722 | 0.5043| 0.4541| 0.4609| 0.6045| 0.6396| 0.7048| 26  |
|                | WordEmbedSynSwap       | 0.3530 | 0.3539| 0.3824| 0.3680| 0.4746| 0.4721| 0.5603| 27  |
|                | WordNetSynSwap         | 0.3488 | 0.3650| 0.3807| 0.3605| 0.4488| 0.4474| 0.5451| 16  |
| ANTIQUE       | original query        | 0.2286 | 0.2169| 0.2175| 0.2065| 0.2663| 0.4212| 0.3338| 200 |
|                | NeighCharSwap         | 0.1559 | 0.1478| 0.1590| 0.1451| 0.1841| 0.2868| 0.2514| 199 |
|                | RandomCharSub         | 0.1623 | 0.1593| 0.1558| 0.1476| 0.1887| 0.2797| 0.2486| 182 |
|                | QWERTYCharSub         | 0.1613 | 0.1527| 0.1602| 0.1550| 0.1922| 0.2978| 0.2664| 197 |
|                | RemoveStopWords       | 0.2270 | 0.2160| 0.2222| 0.2153| 0.2693| 0.3830| 0.3200| 199 |
|                | T5DescToTitle         | 0.1673 | 0.1646| 0.1601| 0.1669| 0.2000| 0.2695| 0.2397| 136 |
|                | RandomOrderSwap       | 0.2286 | 0.2168| 0.2178| 0.1978| 0.2665| 0.4134| 0.3248| 200 |
|                | BackTranslation       | 0.1618 | 0.1546| 0.1602| 0.1438| 0.2036| 0.3045| 0.2584| 93  |
|                | T5QQP                 | 0.2201 | 0.2065| 0.2095| 0.1962| 0.2614| 0.3926| 0.3214| 105 |
|                | WordEmbedSynSwap       | 0.1759 | 0.1719| 0.1902| 0.1690| 0.2142| 0.3245| 0.2828| 124 |
|                | WordNetSynSwap         | 0.1791 | 0.1751| 0.1957| 0.1765| 0.2117| 0.3244| 0.2733| 71  |

of relevant documents on the re-ranking set increases (e.g. BM25 has Recalls @10, @100 and @1000 on average of 0.06, 0.25 and 0.48 for misspelling query variations), BERT still struggles (negative A) with query variations. This indicates that even if we increase the number of relevant documents in the list to be re-ranked, the re-ranking still fail when facing the 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 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 of 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 can not be attributed to unjudged documents being brought up in the ranking by the query variations.

11Similar results are obtained for other neural rankers.
5.1.1 Robustness by Query Variation Category. In order to study the effect of each query variation category, Figure 2 displays the nDCG@10 \( \Delta \) (difference in effectiveness when replacing the original query by its variation) distribution per category and model. Although some queries variations have a positive effect (points with positive \( \Delta \)), the distributions are mostly skewed towards effectiveness decreases (negative \( \Delta \)).

First we see that the on average the decreases are higher for the misspelling category: -0.25 and -0.08 of nDCG@10 \( \Delta \) for TREC-DL-2019 and ANTIQUE respectively. We hypothesize that the effect is higher on TREC-DL-2019 due to it having smaller queries than TREC-DL-2019 (see average number of terms per query on Table 4).

The second highest effect on both datasets are the query variations from the paraphrasing category (-0.08 and -0.03 of nDCG@10 \( \Delta \)) followed by naturality (-0.05 and -0.03). Compared to the misspelling variations which in most cases degrade the effectiveness of our models, paraphrasing and naturality have more queries for which the effect is positive, rendering the overall nDCG@10 \( \Delta \) smaller.

Queries from the ordering category have the least effect (less than 0.01). Since traditional methods are in fact bag-of-words models, changing the word order will not have any effect on them, which makes the average of all models’ nDCG@10 \( \Delta \) closer to zero. In the following section, we take a further look at how each type of ranking model is affected by each query variation method.
5.1.2 Robustness by Model Category. When we consider how different models are affected by the query variations, we see from Figure 2 that the exception of ordering, which has no effect on BM25, RM3 and KNRM, other transformations have a similar overall distribution of nDCG@10 Δ amongst different models. In order to understand if models (and category of models) make mistakes on the same queries, we label the models as follows: BM25 and RM3 are labelled as Trad (lexical matching), KNRM and CKNRM (neural network based) are labelled as NN and EPIC, BERT, T5 are labelled as TNN (transformer language model based). We then represent each model with the nDCG@10 Δ values obtained for each query and variation method resulting in a total of #Q × #M features per model. In order to visualize them we reduce this representation to 2 factors with tSNE [53], as shown in Figure 3. We observe that even though models have similar magnitudes and direction of nDCG@10 Δs, classes of models as indicated by color are clustered indicating that the query variations have similar effects for each type of model. We make a similar observation when we consider the correlation of nDCG@10 Δ amongst models for the different types of transformations as displayed in Table 6, where there are groups with higher similarity that roughly correspond to the different types of models.

While Trad models have decreases of -0.03 (TREC-DL-2019) and -0.01 (ANTIQUE) nDCG@10 for naturality query variations, the effect is higher on TNN: -0.05 and -0.04 respectively. This is evidence that neural ranking models based on heavily pre-trained language models have a slight preference for natural language queries as opposed to keyword queries, which is a finding aligned with previous work [17]. Another interesting finding is that the word order does not have a great effect on TNN models (decreases smaller than 0.01). This is in line with recent research that indicates that the word order might not be as important as initially thought for transformer models [45, 51].

5.2 Fusing Query Variations

Although on average query variations make models less effective, there are cases when there are effectiveness gains (as shown with the positive nDCG@10 Δ in Figure 2). This motivates the combination of different query variations to obtain better ranking effectiveness. In order to understand whether we can improve effectiveness of models by combining different query variations, we compare different methods for combining queries, as displayed in Table 7. RRF_C indicates that we fuse the results obtained from the query variations obtained after applying MC methods using the Reciprocal Rank Fusion (RRF) method [15], and RRF_ALL fuses the results obtained by all query variation methods.12

First we see that there is potential to have significant effectiveness gains, as shown by the last line (best query) where we always use the query with the highest retrieval effectiveness amongst query variations and the original query. The results show that combining query variations with RRF is better than using query variations individually (Table 5), and sometimes it is even the same as using the original query (no statistical difference). Our results indicate that while rank fusion mitigates the decreases in effectiveness of different query variations (RRF_ALL decreases are of 3% and 10% nDCG@10 for TREC-DL-2019 and ANTIQUE respectively when compared to the original query), it does not improve the effectiveness over using the original query.

5.2.1 When are query variations better? To better understand when models benefit from different query variations, we plot the distribution of query variations that improve over the original query by ranking model and query variation category in Figure 4. We see that overall the queries obtained through the naturality and paraphrasing methods are the ones that improve over the original queries the most. Intuitively, paraphrasing query variations can potentially rewrite the query with better terms (e.g. ‘why do criminals practice crime’ → ‘why do criminals practice misdemeanour’ +0.13 nDCG@10 for BERT using WordEmbedSynSwap), make queries grammatically correct (e.g. ‘how sun rises’ → ‘how does the sun rise’ +0.03 nDCG@10 for BERT using T5/QP) and also correct spelling mistakes (e.g. ‘what is sociology’ → ‘what is sociology’ +0.47 nDCG@10 for BERT using BackTranslation). Naturality methods make the queries shorter (e.g. ‘who is robert gray’ → ‘robert gray’)

12ordering was not included in the experiments as a separated row since it only has one method, but it is included in the RRF_ALL method.
Table 7: Effectiveness (nDCG@10) of different methods when employing rank fusion (RRF) of the rankings obtained by using different sets of queries, e.g. RRF\textit{misspelling} fuses queries generated by \textit{misspelling} methods. Bold indicates the highest values observed for each model and ↓↑ subscripts indicate statistically significant losses/improvements, using t-test when compared against the same model with the original queries.

\begin{tabular}{lcccccccc}
 & BM25 & RM3 & KNRM & CKNRM & EPIC & BERT & T5 \\
\hline
\textbf{TREC-DL-2019} & & & & & & & \\
original query & 0.4795 & 0.5156 & 0.5015 & 0.4931 & 0.6240 & 0.6449 & 0.6997 \\
RRF Misspelling & 0.3038\downarrow & 0.3092\downarrow & 0.3233\downarrow & 0.3175\downarrow & 0.3830\downarrow & 0.4164\downarrow & 0.4650\downarrow \\
RRF Naturality & 0.4751 & 0.4972 & 0.4855 & 0.4639 & 0.5901 & 0.6161 & 0.6628 \\
RRF Paraphrasing & 0.4742 & 0.4865 & 0.4805 & 0.4330↓ & 0.5847 & 0.6120 & 0.6629 \\
RRF All & 0.4746 & 0.4976 & 0.5027 & 0.4951 & 0.5902↓ & 0.6031↓ & 0.6453↓ \\
\hline
\textbf{best query} & 0.5401↑ & 0.5774↑ & 0.6055↑ & 0.6121↑ & 0.6990↑ & 0.7194↑ & 0.7595↑ \\
\hline
\textbf{ANTIQUE} & & & & & & & \\
original query & 0.2286 & 0.2169 & 0.2175 & 0.2065 & 0.2663 & 0.4212 & 0.3338 \\
RRF Misspelling & 0.1712↓ & 0.1661↓ & 0.1758↓ & 0.1659↓ & 0.2060↓ & 0.2750↓ & 0.2435↓ \\
RRF Naturality & 0.1847↓ & 0.1866↓ & 0.2029 & 0.2033 & 0.2406↓ & 0.3173↓ & 0.2707↓ \\
RRF Paraphrasing & 0.1909↓ & 0.1845↓ & 0.1917↓ & 0.1762↓ & 0.2380↓ & 0.3394↓ & 0.2882↓ \\
RRF All & 0.1998↓ & 0.1970↓ & 0.2150 & 0.2037 & 0.2430↓ & 0.3179↓ & 0.2729↓ \\
\hline
\textbf{best query} & 0.2716↑ & 0.2682↑ & 0.2985↑ & 0.2843↑ & 0.3370↑ & 0.4488↑ & 0.3925↑ \\
\end{tabular}

+0.34 nDCG@10 for BERT using RemoveStopWords), removing unnecessary information from the original query on certain cases.

6 CONCLUSION

In this work we studied the robustness of ranking models when faced with query variations. We first described a taxonomy of transformations between two queries for the same information need—of the query and \(I, I’, I’’ \) techniques to improve the robustness of existing ranking pipelines.
[62] Shengyao Zhuang and Guido Zuccon. 2021. Dealing with Typos for BERT-based Passage Retrieval and Ranking. arXiv preprint arXiv:2108.12139 (2021).
[63] Guido Zuccon, Joao Palotti, and Allan Hanbury. 2016. Query variations and their effect on comparing information retrieval systems. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. 691–700.