A Quantitative Analysis of Noise Impact on Document Ranking

Edward Giamphy, Kevin Sanchis, Gohar Dashyan, Jean-Loup Guillaume, Ahmed Hamdi, Lilian Sanselme, Antoine Doucet

A Quantitative Analysis of Noise Impact on Document Ranking

1st Edward Giamphy
Preligens and *L3i*
Paris, FRANCE
edward.giamphy@preligens.com

2nd Kevin Sanchis
Preligens, Paris, FRANCE
kevin.sanchis@preligens.com

3rd Gohar Dashyan
Preligens, Paris, FRANCE
gohar.dashyan@preligens.com

4th Jean-Loup Guillaume
L3i, *La Rochelle University*
La Rochelle, FRANCE
jean-loup.guillaume@univ-lr.fr

5th Ahmed Hamdi
L3i, *La Rochelle University*
La Rochelle, FRANCE
ahmed.hamdi@univ-lr.fr

6th Lilian Sanselme
Preligens, Paris, FRANCE
lilian.sanselme@preligens.com

7th Antoine Doucet
L3i, *La Rochelle University*
La Rochelle, FRANCE
antoine.doucet@univ-lr.fr

***Abstract*—After decades of massive digitization, a substantial amount of documents exists in digital form. The accessibility of these documents is strongly impacted by the quality of document indexing. Most of these documents are indexed in noisy versions that include numerous errors. The noise can be due to manual input mistakes or optical character recognition process and results in errors like spelling mistakes, missing characters, and others. This paper presents a study of the impact of noise on document ranking, an essential task in natural language processing (NLP) with wide-ranging practical applications. We provide a deep and quantitative analysis of the impact of recognition errors on document ranking by testing two popular ranking models on several noisy versions of a subset of the MS MARCO passage ranking dataset, with various levels and types of noise. Our study provides insights into the challenges of document ranking under noisy conditions and advocates for developing ranking models that are more robust to noise.**

***Index Terms*—Information Retrieval, Document Ranking, Indexing, Noise, OCR Errors, Natural Language Processing.**

I. Introduction

Due to recent technological advances and the growing availability of digital platforms, a substantial amount of documents exists in digital form. Efforts of cultural heritage institutions toward digitization are increasingly contributing to the production of massive amounts of digitized documents. These documents are transcribed either manually using human efforts or automatically via optical character recognition (OCR) or automatic speech recognition systems (ASR) depending on their format. As a result, their textual content is always noisy due to human typing mistakes, OCR/ASR errors or other factors. Yet, the accessibility of these documents is correlated to the way they are indexed, and it is precisely these error-prone versions, where some words are erroneous, that are used during indexing by search engines. This represents a serious problem for document indexing and, subsequently, for document retrieval [1], [2].

Ranking is a crucial task in information retrieval, with a wide range of real-world applications, such as question answering and text summarization. In these applications, the objective is to automatically sort a set of documents based on their relevance to a given query. Ranking allows users to quickly and efficiently find the most relevant documents from a large corpus of text. However, in real-world scenarios, the performance of ranking models can be significantly affected by the different types of noise [3]. It is therefore essential to understand the impact of noise to develop more robust models.

While previous works were conducted to evaluate the impact of OCR errors [4], few works focus on broader noise types that can be induced by other sources. Hence, in this work, we investigate the impact of different types of noise on ranking, specifically insertion, deletion and substitution, in order to better understand the robustness of document ranking models under noisy conditions. The results of this work can support the development of more accurate practical applications, such as web search engines and information retrieval systems with noisy text. To the best of our knowledge, this is the first time that a study explores the impact of noise with such fine granularity. By open-sourcing our noisy datasets we also hope to contribute to the development of NLP models that can handle noisy inputs effectively.

In this study, we focus on the performance of two popular ranking models, BM25 [5], a probabilistic ranking function, and DistilBERT [6], a Transformer-based model. We use various metrics to evaluate the performance of these models under different levels and types of noise.

More precisely, our work brings three main contributions:

- We give an in-depth and quantitative evaluation of the impact of spelling errors on the performance of two popular ranking methods.
- We analyse the effect of different types and distributions of noise.
- We create and provide several datasets with different levels and types of noise, to support research efforts to develop ranking models more robust to noisy text inputs.

The rest of this paper is structured as follows. Section II introduces related works on document ranking and dealing with noise in text. Section III describes the experimental protocol followed to study the impact of noise and provides

technical details on experimental aspects. Section IV presents the performance of the document ranking models under noise constraints and discusses the obtained results. Finally, Section V concludes the paper with a discussion of several open problems and possible research directions.

II. Related works

a) Impact of noisy inputs on downstream NLP tasks performance: Many NLP works have focused on processing noisy data for sentence boundary detection, tokenization and part-of-speech tagging [7], [8]. Some other works have evaluated the effects of noisy texts on other NLP tasks such as text categorization [9], [10], document summarization [11], machine translation [12], named entity recognition [13], entity linking [14] and topic modeling [15]. Van Strien et al. [16], for instance, showed that processing low-quality documents impairs the performance of six NLP tasks including sentence segmentation and dependency parsing. Hamdi et al. [17] showed that the performance of named entity recognition systems can have a significant drop of F1-score from 90% to 50% for character error rates between 2% and 30%.

b) Document Ranking: Information Retrieval (IR) is the task of retrieving relevant documents from a large collection of documents in response to a user's query. The retrieved documents are generally ranked with respect to some relevant notion. Ranking is used in many real-world applications, such as search engines, recommender systems or document retrieval systems [18], [19]. Existing approaches for ranking include traditional probabilistic models such as BM25 [5], which rely on sparse representation of the queries and documents. With the advent of deep learning, more advanced models such as Transformer-based models like BERT and DistilBERT [6], [20] have been proposed. These models use dense representation of the documents and queries and have shown state-of-the-art performance in various NLP tasks, including document ranking. More recent works have also explored the use of pre-trained language models such as T5 and UniLM [21], [22] for document ranking, resulting in significant performance improvements. In this work, we focus on sparse and dense representation models in a noisy context.

c) Impact of noisy inputs on information retrieval: Many works have focused on IR from noisy data [23]. The impact of errors in documents is well studied in research tasks [24]. Chiron and al. [2], for instance, proposed a method to estimate the risk that a user's query might fail to match with the noisy documents stored in digital libraries. Taghva et al. [4] showed that moderate OCR error rates do not have a high impact on the effectiveness of classical IR measures. A more recent work showed that information retrieval can be damaged with only 5% of error rate at the character-level [25]. Interestingly, de Oliveira et al. [26] showed that, with comparable error rates in the documents, longer ones are more impacted by OCR errors than shorter ones.

Our work is in the same spirit as that of Bazzo et al. [25]: we quantitatively evaluate the impact of noise on document ranking. However, contrary to them, we discuss in further detail the impact of the different types of noise-inducing operations (insertion, substitution, deletion) with different levels of error density.

III. Methodology

In this section, we present the methodology of our work that explores noise impact on ranking models and more precisely on passage ranking. Passage ranking is an essential research task of IR that involves retrieving passages from a large corpus of text that are relevant to a given query and ranking these passages by relevancy. This task has several challenges including identifying relevant passages in long documents, handling the variability in language usage and writing styles, or handling ambiguous queries and more generally any form of noise.

We hereby focus on noise-related challenges, but noisy datasets are a scarce resource and none of the available ones meet our needs. Therefore, we create synthetic datasets simulating different types of noise. Creating our own noisy datasets allows us to fully control the distribution and intensity of the noise, which is crucial for a comprehensive evaluation of ranking models under noisy conditions.

In a nutshell our methodology is as follows:

- We operate on the TREC 2020 deep learning track dataset.
- We inject noise of different types and levels in the passages of that corpus using the `nlpaug` library, and thus obtain 72 different noisy versions of the dataset.
- We apply two ranking models, BM25 and DistilBERT, to rank passages on these noisy datasets.
- We use several metrics, precisely NDCG, Recall@1000, MAP and MRR, to evaluate the performance of these models under noisy conditions.

A. Dataset and passage ranking task

We conduct our study on the MS MARCO passage ranking dataset [27] as it has gained popularity in the IR community these recent years and acts as a reference. The MS MARCO passage ranking dataset is a collection [28] comprising a passage corpus, test queries and test relevance judgments. The dataset includes a corpus of 8.8M passages gathered from Bing search engine results. The length of the passages varies and the mean length of each passage is 56 tokens (median: 50, max: 362). More details on the passage lengths distribution are available in [29].

The MS MARCO passage ranking dataset is split into 3 sub datasets: a training, a development and a test set, each of which consists of pairs of queries and relevant passages. Each query has one relevant passage on average.

For our experiments, we use the MS-Marco-passagetest2020-top1000 set [29]. It consists of an initial ranking of 1000 passages per test query. We consider the union of these candidate passages and focus on the full ranking task on the resulting corpus. The reason for that is the heavy quantity of noising and ranking runs to be performed, which requires a corpus that is smaller than the

official full-ranking one, but significant enough to be relevant. This dataset contains $\simeq$ 170k passages. Note that the task we perform is different from the full-ranking task where the corpus is much larger. It is also different from the re-ranking task where only the 1000 candidates per query are considered for a given query, and not the union of all candidates.

B. Noise injection

Noise injection mostly depends on three parameters: the type of noise, the amount of noise quantified at a character level, and how it is distributed in words.

To measure the amount of noise injected in text, we use the Character Error Rate (CER) that is a common metric used to evaluate the performance of speech recognition or machine translation systems. In the context of our study, CER compares, for a given degraded text, the total number of characters, including spaces, to the minimum number of insertions, substitutions and deletions of characters required to obtain the clean text. For a given CER, the distribution of errors can be uniform in the words or, on the contrary, concentrated in some words. We therefore also use the Word Error Rate (WER) that is defined in the same way at the word level. For a given CER, we can have a few heavily modified words (hence a low WER) or many slightly modified words (hence a high WER). Note that in our study, when used to measure the noise in a given dataset, the term CER (resp. WER) refers to the mean CER (resp. WER) over the corpus' passages.

We focus on three types of character-level noise – insertion, deletion, and substitution – which are commonly encountered in real-world scenarios and have been shown to have a significant impact on the performance of NLP models [30]. By focusing on noise types, we can explore any *nature* of noise, e.g. random, OCR or keyboard-induced noise, regardless of the source, as any nature of noise can be reconstructed as a combination of character-level insertions, deletions, and substitutions. Hence, focusing on the noise type rather than its nature allows us to investigate the impact of noise in a more comprehensive and generic manner, without being tied to a specific scenario.

Following this idea, we inject random character-level errors in the text: insertion corresponds to a random injection of a new character, deletion is when a character is removed randomly and substitution is the replacement of a character (in a word) by another random character. As mentioned above, although some more realistic injections might be considered, for example by injecting keyboard-related errors (a character will be more likely replaced by a character close on a keyboard), or OCR-related errors (a character will be replaced by a character with a similar shape), we argue that, in the context of this study, changing the nature of injected errors would not make a significant difference, as we focus on the impact of varying amounts and distributions of noise in a given text, rather than tackling specific real-world scenarios. The choice of working with randomly injected errors is therefore natural as it is the most generic way of creating noise.

We inject each of the three types of noise with different intensities by varying the CER from 0% to 36% with intervals of 3% as described in Figure 2. For each value of CER, we also study two different regimes of error distribution: one where errors are distributed in few words in the text, but where affected words are severely modified (low WER), that we refer to as *Batch 1*, and one where errors are more evenly spread out between words (higher WER) that we refer as *Batch 2*. In table I, we show an example of passage and its degraded versions for batches 1 and 2 and for the three types of noise. Figure 1 provides the characteristics of each noisy dataset we produced. We can see that the WER and CER are rather similar for the three types of noise. We also note that the measured WER in Batch 2 is about twice higher than in Batch 1 at equal measured CER. This is consistent with the fact that, in Batch 2, we inject noise that is more uniform and therefore strongly impacts the WER, at equal CER. For insertion and substitution, the WER in Batch 2 can even exceed 1 (it appears when the number of modifications exceeds the total number of words), even if the CER is kept below 0.4.

To synthesize, we inject noise with three noise types, two different distributions and an increasing CER intensity taking 12 different values. In total, we therefore generate 72 noisy versions (2 batches x 3 noise types x 12 target CER) of the MS MARCO Passage Ranking test collection. Data produced and used in this work are available by following this link[1].

C. Noise injection tool

To simulate the chosen noise types in our study, we use the `nlpaug` library [31], a popular open-source NLP augmentation library that provides various functions for data augmentation. This library offers a wide range of augmentation techniques that can be easily integrated into our experimental pipeline.

However, `nlpaug` was not specifically designed for noise injection, and therefore, may have some limitations in terms of accuracy and relevance of the simulated noise. For example, we found that obtaining the desired level of noise using `nlpaug` was often challenging and resulted in inaccuracies in the target CER and WER. To overcome this issue, we tolerate a slight error margin with the corpus mean CER as a reference point. Specifically, for each noisy dataset, we aim at generating noise with an average CER that is relatively close to the target CER, such that the absolute difference between the two values does not exceed 0.75. For example if we target a CER of 12%, `nlpaug` might generate a CER of 11.8%. In this case we keep the noisy dataset since we tolerate any value between 11.25% to 12.75%. The noisy datasets have been obtained using a grid search on `nlpaug` parameters and only the most relevant datasets, in terms of CER, have been kept. We illustrate this

[1]https://www.dropbox.com/scl/fo/7mvun4nh3et3ak6bbcrz1/h?dl=0&rlkey=tgw7lrt5kglrdvehpu461ghq3
This link is temporary, it is provided to reviewers only, and is not to be used for any purpose other than evaluating the contributions in this paper. Should the paper be accepted, all the datasets we produced will be moved to an open data repository such as *Zenodo* and made publicly available.

TABLE I
AN EXAMPLE OF NOISE INJECTION AT CER ($\simeq 0.10 - 0.15$) FOR BATCH 1 AND BATCH 2

Original passage		The definition of expressed is to convey by words, gestures or conduct. You can learn more about the definition of the word expressed at the Dictionary website.
Batch 1	**insertion**	The definition of expressed is to convey by nw9oKrrdNs, gestures or kcGogn%d8uBcFt. #YRoTu can learn more about the definition of the word expressed at the tDmiGc%tJidofnGaQrfy website.
	deletion	The definition of expressed to convey by words, gestures or conduct. You can learn more about definition of word expressed at the Dictionary.
	substitution	The O3_C_WS(M3 of expressed is to convey by words, gestures or conduct. You can learn more about the definition &D 5c& %E5y expressed at 7kA Dictionary website.
Batch 2	**insertion**	The d0ef(initsion of expressed is Hto convey b_y owoxrds, gestures or cFonducst. YJou cZan lxearjn more about tMhe definition #of the word expressed aKt thZe Dictionary website.
	deletion	Th definition of expreed is to onve by wors, gestures or coduc. You cn lern ore abou th definition o the wrd exprssd a he Dictionary wsite
	substitution	Jhe definition Gf expressed ik to c#Qvey by 6ords, gestures or connucg R You Can leaEn more abeut the dQfinmtion of whe word expressed at thO Dictio$ar6 wessiXe.

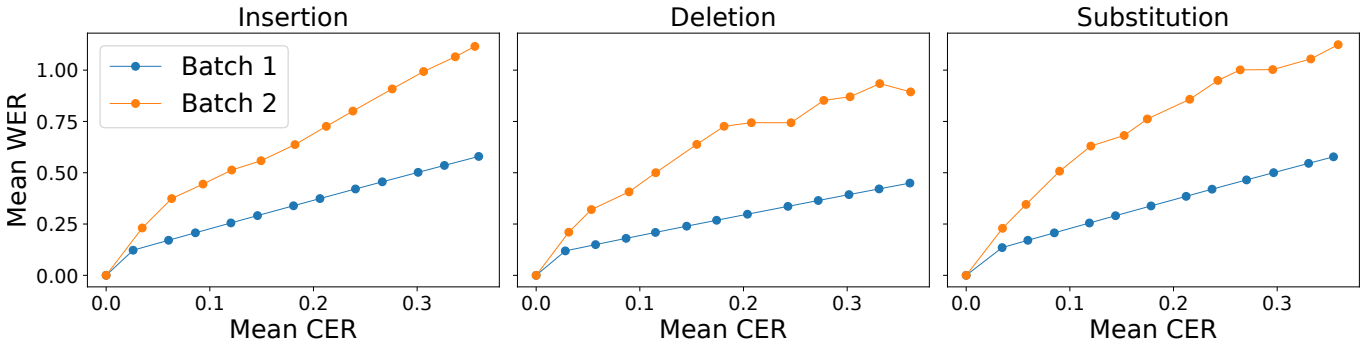

Fig. 1. Noise distribution in Batch 1 and Batch 2 describing the proportion of WER as a function of the CER of the noisy datasets. Every point in the figure corresponds to a noisy dataset, except the point at 0 CER and WER which is our original corpus.

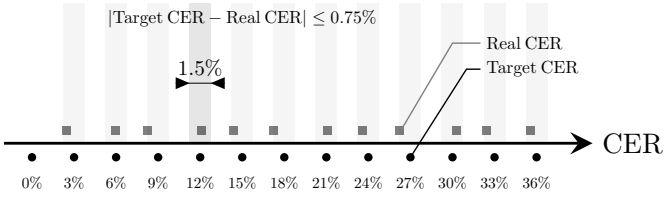

Fig. 2. Schematic showing the obtained mean CER of the noisy corpus against the initially targeted CER values. The tolerance intervals are shown in grey and white bands.

margin in Figure 2. It is to be noted that the curves in Figure 1 and all subsequent discussions in this article always refer to the real CER of the noisy datasets and not the target CER.

Despite these limitations, we believe that `nlpaug` is a suitable choice for our study, as it provides a convenient and efficient way to simulate different types of noise and allows us to obtain meaningful levels of noise intensity in our datasets. Moreover, it has been widely used in previous studies [32], [33] and has demonstrated its effectiveness in improving the performance of NLP models in some contexts [34]. !

D. Ranking models

The aim of this study is to evaluate the performance of document ranking models on the aforementioned noisy datasets. We use two classical ones, namely BM25 [5] and DistilBERT [6] on the passage ranking task. We use the `Pyserini` python toolkit to run our experiments [35].

BM25 is a well-known model in the field of information retrieval that has been widely used for various retrieval tasks, such as web search and document ranking. BM25 is a sparse representation model that computes a score based on the frequency of the query terms in the document, as well as their inverse document frequency. On the other hand, DistilBERT is a state-of-the-art language model that uses a dense representation approach based on deep neural networks. DistilBERT is a pre-trained model that can be fine-tuned for various NLP tasks, including passage ranking. Precisely, it is a distilled version [36] of BERT [20], retaining 97% performance but being 60% faster and using only half the number of parameters. Unlike BM25, DistilBERT generates dense representations that encode semantic and syntactic information that can capture complex relationships between words and phrases.

Sparse and dense representations diverge in the way they encode the queries and documents. As suggested by their name, sparse (resp. dense) representation models encode the documents as sparse (resp. dense) vectors. By comparing these two types of models in a noisy context, we can gain insight into the strengths and weaknesses of sparse and dense representation models, and how they perform under different levels and types of noise. This comparison is particularly interesting because the two models use fundamentally different approaches to represent text. Moreover, we believe that BM25 and DistilBERT are representative of their family of models and we would expect similar results if we conducted experiments with other models.

E. Evaluation in noisy conditions

To evaluate the performance of passage ranking models under noisy conditions, we focus on several commonly used metrics in information retrieval. These metrics evaluate the relevance of retrieved passages based on judgments provided by human assessors. We use the normalized discounted cumulative gain (NDCG) [37], Recall@1000, the mean average precision (MAP) [38] and the mean reciprocal rank (MRR) [39]. NDCG measures the effectiveness of a ranking model by taking into account the relevance of retrieved documents at different positions. Recall@1000, on the other hand, measures the proportion of relevant documents that are retrieved by a model within the top 1000 documents. MAP measures the mean of the average precision of a model over every query. Finally, MRR is the mean, over every query, of the inverse of the rank of the first relevant document retrieved by a model.

These metrics provide comprehensive and insightful information on the performance of passage ranking models. We implemented these metrics with the official evaluation procedure of the TREC 2020 deep learning track with NIST labels [29]. Note that we ensure our implementation of BM25 and DistilBERT indexed on the full MS MARCO passage ranking corpus obtains the same performance as in the `Pyserini two-click reproduction matrix` [2].

IV. Results and discussion

We apply BM25 and DistilBERT on the generated noisy datasets. Results for all combinations of datasets and batches can be found in Figure 3. As expected, the ranking performance decreases substantially as the level of CER increases. Nevertheless, we sometimes observe a slight improvement. This phenomenon may be due to disturbances in the performance measurements and highlights that variance studies should ideally be performed.

Our results indicate that the amount of injected noise in the text has an impact on all metrics used to evaluate the ranking performance. We can notice that Recall@1000 and MRR are only slightly affected by the noise in Batch 1. However, these metrics are significantly more degraded in Batch 2.

We found that the three types of noise have a similar effect on ranking performance, although there were some observed differences depending on the distribution of errors in the text. Specifically, insertion was slightly more detrimental than other noise types in Batch 1. Paradoxically, insertion was less detrimental to ranking performance on Batch 2. We deem that these observations are not significant due to the very small difference between the three types in terms of performance.

Consequently, we consider that BM25 and DistilBERT behave comparably in the presence of character-level insertion, deletion, or substitution errors which suggests that the noise type does not have a significant impact on the evaluated ranking models. Instead, it is the overall quantity and distribution of noise that affects the performance.

The ranking performance comparison over Batch 1 and Batch 2 allows us to draw general conclusions on the sensibility of document ranking models with noise distribution in the text. The differences observed between Batch 1 and Batch 2 indicate that document ranking performance in noisy conditions is strongly influenced by the distribution of character-level errors in the words of the text. Specifically, we observe that when errors are heavily concentrated in a few words, the impact on retrieval performance is less pronounced compared to when errors are evenly distributed across multiple words. This finding is particularly relevant as, to the best of our knowledge, this is the first time that a study explores the impact of noise distribution on ranking performance. We believe that this is due to the tokenization process on which both methods depend: a single heavily degraded word in a passage would have a small effect on the vocabulary, while many slightly degraded words would have a large impact, by creating a high amount of noisy tokens in the vocabulary. Consequently, the main factor of performance degradation seems to be the WER in the datasets.

The dependence on tokenization of BM25 and DistilBERT would explain why both models exhibit similar behavior in response to the noise injected despite the fact that they use different approaches to ranking. In fact, in Batch 2, when the Mean CER exceeds 0.2, the two ranking methods perform comparably regardless of the quantity, type and distribution of errors. It is interesting to note that BM25 and DistilBERT have very close Recall@1000. Moreover, in Batch 2, in terms of MAP and NDCG, models' performance get closer as the CER increases and in some cases, BM25 even outperforms DistilBERT. This tends to show that DistilBERT is more strongly impacted by high levels of noise than BM25. These observations are informative as they highlight that dense representation models do not show any particular robustness to noise compared to sparse representation models. Hence, the development of models adapted to the presence of noise in the text is needed to mitigate the detrimental impact on ranking performance.

V. Conclusion

In this work, we explored the impact of noise on document ranking using two well-known models, BM25 and DistilBERT, on the MS MARCO passage ranking test set. We injected noise into this dataset with three types of character-level errors.

[2]https://castorini.github.io/pyserini/2cr/msmarco-v1-passage.html

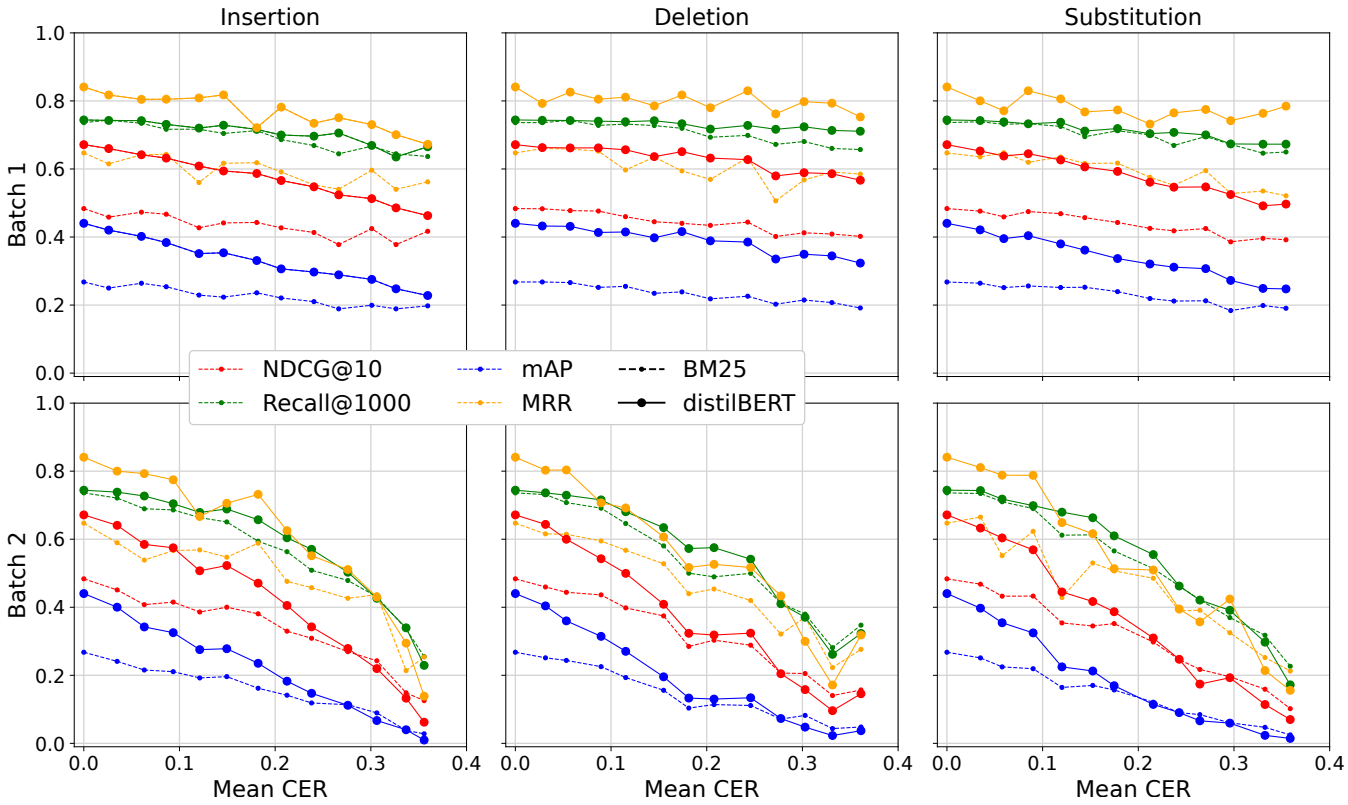

Fig. 3. Performance of BM25 and DistilBERT with an increasing amount of CER in the injected noise in the MS MARCO passage ranking test set. The different ranking methods are represented in two different marker types and line-styles, the different metrics are shown in different colors.

We defined two regimes of noise injection, one with a low WER and a high number of noisy characters per word (Batch 1), and the other with a high WER and a low number of noisy characters per word (Batch 2). We led a comprehensive evaluation of passage ranking models using multiple metrics: NDCG, Recall@1000, MRR, and MAP.

Our results indicate that, as expected, the retrieval performance decreases substantially with the level of CER. We observe that the performance of BM25 and DistilBERT is affected similarly by different types of noise, suggesting that neither sparse nor dense representation models are particularly suited for ranking under noisy conditions. We believe these observations are due to the tokenization process of these types of model. Additionally, our findings show that ranking performance in noisy conditions is dependent on the distribution of character-level errors in the words of the text, which is well represented by the WER associated with our level of CER: for a given CER value, the higher the WER is, the lower the ranking performance will be.

Moreover, by open sourcing the noisy datasets produced for this study, we believe that our work will be valuable for researchers and practitioners working on NLP tasks that involve noisy input data.

Future work could explore the impact of mixing different types of noise on document ranking, or study other natures of noise such as OCR-related or keyboard-related. Additionally, it would be interesting to investigate how document length impacts the performance of ranking in noisy conditions as the dataset we used contains relatively short text documents. Finally, our results suggest that neither sparse nor dense representation models are superior to the other in the context of noisy ranking. Dense representation models are even more degraded although they perform better for a reasonable quantity of CER. Hence, further research is required to determine which type of representation is most effective in dealing with text containing spelling errors. An interesting track to investigate this matter would be to explore the behavior of different tokenizers in a noisy context.

ACKNOWLEDGMENT

The authors would like to express their gratitude to Tugdual Ceillier, Vincent Vidal, Marie-Caroline Corbineau and Emanuela Boros for their valuable contributions in insightful discussions about this work. Their inputs have enriched our work and helped us to improve our methodology and analysis. We acknowledge and appreciate their support and involvement in this research project.

REFERENCES

[1] D. Grangier, A. Vinciarelli, and H. Bourlard, “Information retrieval on noisy text,” IDIAP, Tech. Rep., 2003.

[2] G. Chiron, A. Doucet, M. Coustaty, M. Visani, and J.-P. Moreux, "Impact of ocr errors on the use of digital libraries: towards a better access to information," in *JCDL*, 2017.

[3] K. Taghva, J. Borsack, A. Condit, and S. Erva, "The effects of noisy data on text retrieval," *JASIST*, 1994.

[4] K. Taghva, J. Borsack, and A. Condit, "Effects of ocr errors on ranking and feedback using the vector space model," *IPM*, 1996.

[5] S. E. Robertson and K. S. Jones, "Relevance weighting of search terms," *JASIST*, 1976.

[6] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," 2020.

[7] D. Lopresti, "Performance evaluation for text processing of noisy inputs," in *Proceedings of the 2005 ACM SAC*, 2005.

[8] ——, "Measuring the impact of character recognition errors on downstream text analysis," in *SPIE*, 2008.

[9] D. J. Ittner, D. D. Lewis, and D. D. Ahn, "Text categorization of low quality images," in *SDAIR*. Citeseer, 1995.

[10] G. Zu, M. Murata, W. Ohyama, T. Wakabayashi, and F. Kimura, "The impact of ocr accuracy on automatic text classification," in *Content Computing*, C.-H. Chi and K.-Y. Lam, Eds., 2004.

[11] H. Jing, D. Lopresti, and C. Shih, "Summarizing noisy documents," in *SDIUT*, 2003.

[12] A.-O. Yaser, "Effect of degraded input on statistical machine translation," in *SDIUT*, 2005.

[13] A. Hamdi, A. Jean-Caurant, N. Sidère, M. Coustaty, and A. Doucet, "Assessing and minimizing the impact of ocr quality on named entity recognition," in *TPDL*, 2020.

[14] E. Linhares Pontes, A. Hamdi, N. Sidere, and A. Doucet, "Impact of ocr quality on named entity linking," in *ICADL*, 2019.

[15] S. Mutuvi, A. Doucet, M. Odeo, and A. Jatowt, "Evaluating the impact of ocr errors on topic modeling," in *ICADL*, 2018, pp. 3–14.

[16] D. Van Strien, K. Beelen, M. C. Ardanuy, K. Hosseini, B. McGillivray, and G. Colavizza, "Assessing the impact of ocr quality on downstream nlp tasks," 2020.

[17] A. Hamdi, E. L. Pontes, N. Sidere, M. Coustaty, and A. Doucet, "In-depth analysis of the impact of ocr errors on named entity recognition and linking," *Natural Language Engineering*, 2023.

[18] G.Sudeepthi, G. Anuradha, and M. S. P. Babu, "A survey on semantic web search engine," 2012.

[19] Y. Peng, "A survey on modern recommendation system based on big data," 2022.

[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[21] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," 2020.

[22] L. Dong, N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou, and H.-W. Hon, "Unified language model pre-training for natural language understanding and generation," 2019.

[23] W. Croft, S. Harding, K. Taghva, and J. Borsack, "An evaluation of information retrieval accuracy with simulated ocr output," in *SDAIR*, 1994.

[24] M. C. Traub, J. Van Ossenbruggen, and L. Hardman, "Impact analysis of ocr quality on research tasks in digital archives," in *TPDL*. Springer, 2015.

[25] G. T. Bazzo, G. A. Lorentz, D. Suarez Vargas, and V. P. Moreira, "Assessing the impact of ocr errors in information retrieval," in *ECIR*, 2020.

[26] L. L. de Oliveira, D. S. Vargas, A. M. A. Alexandre, F. C. Cordeiro, D. d. S. M. Gomes, M. d. C. Rodrigues, R. K. Romeu, and V. P. Moreira, "Evaluating and mitigating the impact of ocr errors on information retrieval," *International Journal on Digital Libraries*, 2023.

[27] P. Bajaj, D. Campos, N. Craswell, L. Deng, J. Gao, X. Liu, R. Majumder, A. McNamara, B. Mitra, T. Nguyen, M. Rosenberg, X. Song, A. Stoica, S. Tiwary, and T. Wang, "Ms marco: A human generated machine reading comprehension dataset," 2018.

[28] N. Stokes, "TREC: Experiment and Evaluation in Information Retrieval," *Computational Linguistics*.

[29] N. Craswell, B. Mitra, E. Yilmaz, and D. Campos, "Overview of the trec 2020 deep learning track," 2021.

[30] K. Al Sharou, Z. Li, and L. Specia, "Towards a better understanding of noise in natural language processing," in *RANLP 2021*, 2021.

[31] E. Ma, "Nlp augmentation," https://github.com/makcedward/nlpaug, 2019.

[32] V. Awatramani and A. Kumar, "Linguist geeks on WNUT-2020 task 2: COVID-19 informative tweet identification using progressive trained language models and data augmentation," in *W-NUT 2020*, 2020.

[33] J. Novikova, "Robustness and sensitivity of BERT models predicting Alzheimer's disease from text," in *W-NUT 2021*, 2021.

[34] A. Deng and E. Shrestha, "Bert-based transfer learning with synonym augmentation for question answering."

[35] J. Lin, X. Ma, S.-C. Lin, J.-H. Yang, R. Pradeep, and R. Nogueira, "Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations," in *SIGIR 2021*, 2021.

[36] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015.

[37] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of ir techniques," *ACM Trans. Inf. Syst.*, 2002.

[38] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*, 2008.

[39] N. Craswell, *Mean Reciprocal Rank*, 2009.