NevIR: Negation in Neural Information Retrieval

Orion Weller, Dawn Lawrie, Benjamin Van Durme
Johns Hopkins University
oweller@cs.jhu.edu

Abstract

Negation is a common everyday phenomena and has been a consistent area of weakness for language models (LMs). Although the Information Retrieval (IR) community has adopted LMs as the backbone of modern IR architectures, there has been little to no research in understanding how negation impacts neural IR. We therefore construct a straightforward benchmark on this theme: asking IR models to rank two documents that differ only by negation. We show that the results vary widely according to the type of IR architecture: cross-encoders perform best, followed by late-interaction models, and in last place are bi-encoder and sparse neural architectures. We find that most information retrieval models (including SOTA ones) do not consider negation, performing the same or worse than a random ranking. We show that although the obvious approach of continued fine-tuning on a dataset of contrastive documents containing negations increases performance (as does model size), there is still a large gap between machine and human performance.[1]

1 Introduction

Recent work in natural language processing (NLP) has shown that language models (LMs) struggle to understand text containing negations (Ravichander et al., 2022; McKenzie et al., 2022) and have poor performance compared to humans. This unresolved problem has downstream implications for information retrieval (IR) models, which use LMs as the starting backbone of their architectures.

However, work on negation in IR has mainly focused on pre-neural (e.g. no LM) retrieval (Kim and Kim, 1990; McQuire and Eastman, 1998; Averbuch et al., 2004; Kim et al., 2019), with no research into how negation affects modern neural IR. This failure to understand negation in IR can lead to devastating consequences in high stakes situations, like the prominent case where Google Search told users what to do during a seizure by listing off bullet points from a website that was specifically specifying what **not** to do (Figure 1). One can easily imagine other serious failure cases in high-stakes domains such as law, education, or politics. Even for casual everyday usage, a lack of understanding of negation by neural IR ignores an entire category of user queries, such as "Where should I not stay in [vacation town]?", "Who did not win an Oscar in 2023?", or "What information has OpenAI failed to release about GPT-4?"

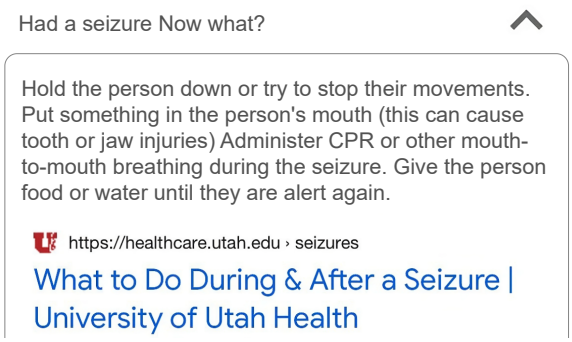

Figure 1: Negation is something not well understood by IR systems. This screenshot shows Google Search making a deadly recommendation because of its failure to catch the negation in the article (e.g. "do not ...").

We aim to fill this gap in the literature by providing a benchmark for Negation EValuation in Information Retrieval, dubbed NevIR (pronounced "*never*"). NevIR builds off of existing work in negation (Ravichander et al., 2022) by using 2,556 instances of contrastive document pairs that differ only with respect to a crucial negation. We then crowdsource query annotations for the two documents in each pair, where each query is only relevant to one of the respective documents and is irrelevant to the other document (Figure 2). By doing so, we can test whether models correctly rank the documents when accounting for the negation.

We find that nearly all IR systems ignore the negation, generally scoring one document of the

[1]Code and data are available at https://github.com/orionw/NevIR

Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 2274–2287
March 17-22, 2024

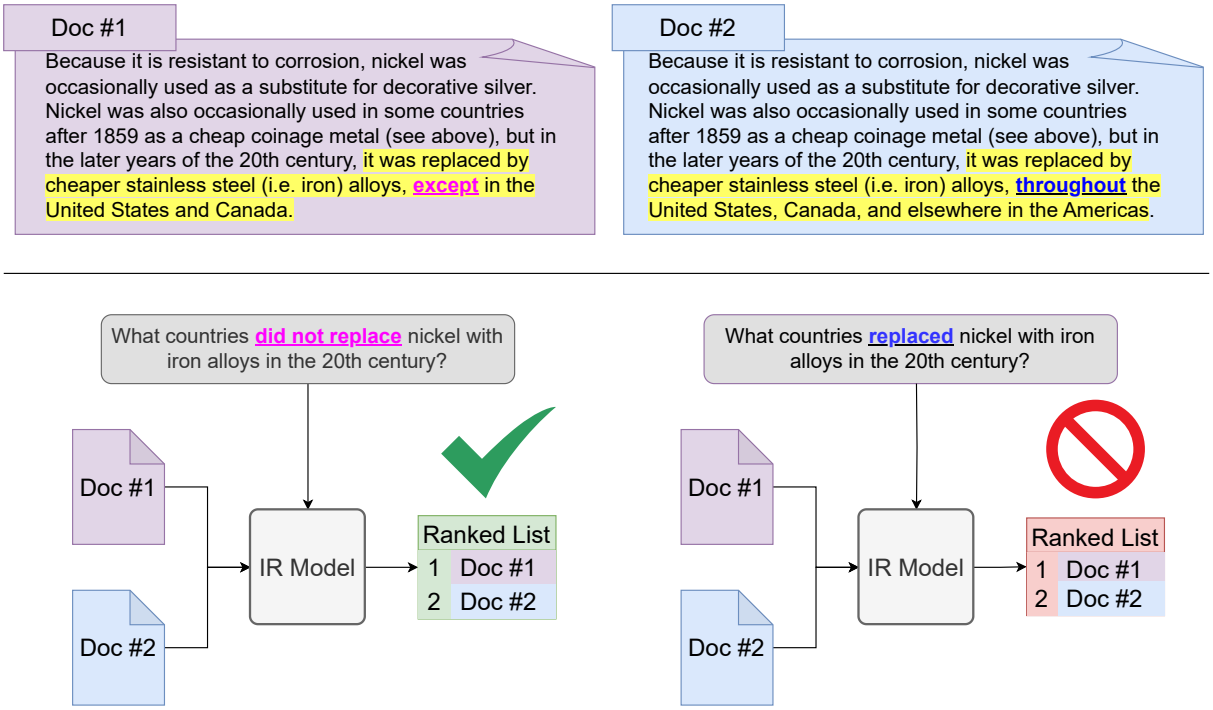

Figure 2: An example instance and the evaluation process. The initial documents from CondaQA (Ravichander et al., 2022) are used to create the queries via Mechanical Turk. The lower half shows the pairwise accuracy evaluation process, where the model must rank both queries correctly. In this example, the IR model scored zero paired accuracy, ranking Doc #1 above Doc #2 in both queries (and failing to take into account the negation).

two higher for both queries. Furthermore, state-of-the-art models perform nearly the same or much worse than randomly ranking the document pairs. We provide analysis of these results, showing that bi-encoder representations of the two documents are nearly identical despite negation words and that late-interaction models such as ColBERT ignore negation words in the MaxSim operator.

We also show that continued fine-tuning of IR models on negation data provides some gains on NevIR, but still leaves significant room to improve (while also slightly hurting performance on traditional benchmarks such as MSMarco). We hope that our analysis will spur increased attention to the problem of negation in information retrieval and provide a dataset for IR training and evaluation.

2 Background

2.1 Motivation

Information Retrieval (IR) is a broadly defined task of finding relevant pieces of information based on a query in natural language. The specifics of IR can vary broadly across languages, domains (e.g. legal), and purposes (e.g. counterarguments, lists, general factoids). Note that many of these specialized cases would be improved through a better understanding of negation, such as lists, counterarguments, and domain-specific language (e.g. legal or medical).

Along with the improvement from neural IR, there has been a surge of interest in retrieval-augmented language models, such as RAG (Lewis et al., 2020), FiD (Izacard and Grave, 2021), and SeeKeR (Shuster et al., 2022). In just the last few months, generative retrieval has been productionized, with systems such as Google's Bard, Bing Chat, and You.com.[2] These systems combine IR models with large language models, enabling them to find and generate responses to queries on the fly.

Thus, as LMs and IR systems become more intertwined and used in production, understanding and improving their failure cases (such as negation) becomes crucial for both companies and users.

2.2 Neural IR

Since 2020, neural models for information retrieval have generally outperformed traditional sparse methods (such as BM25) in most situations (Karpukhin et al., 2020; Khattab and Zaharia, 2020). Given a large collection of training data, these models are optimized using a contrastive loss in order to learn how documents are related to a

[2]https://bard.google.com/, https://www.bing.com/new, and https://you.com

given query. These methods provide several advantages over sparse methods, including the ability to go beyond simple lexical matches to encode the semantic similarity of the natural language text.

Recent work has focused on the ability of neural models to generalize to new domains, without any domain-specific training data (e.g. zero-shot). One prominent benchmark for this type of work is the BEIR dataset suite (Thakur et al., 2021) which evaluates models' generalization on a range of diverse IR datasets. Our work provides both zero-shot (no model fine-tuning) and standard train/test splits to accommodate both paradigms.

2.3 Negation in NLP

Negation has also been an area where LMs typically perform below average (Li and Huang, 2009; He et al., 2017; Hartmann et al., 2021; Ettinger, 2020). Recent work on negation in NLP has shown that although LMs struggle with negation, it does improve with model scaling and improved prompting techniques (McKenzie et al., 2022; Wei et al., 2022). Despite scale improvements, these works (and other follow up works, c.f. Ravichander et al. (2022); Hossain et al. (2022)) have shown that LMs still struggle with negation and are in need of new datasets and methods to improve performance.

As modern IR models use LMs as the backbone of their architectures, it is intuitive that negation will pose problems to IR systems as well. This problem is compounded as IR models are not able to scale to larger LMs as easily, due to efficiency and latency constraints on processing large amounts of documents in real-time.

2.4 Negation in IR

Negation has been a weak point for information retrieval methods throughout the years. Early work in information retrieval (Kim and Kim, 1990; Strzalkowski et al., 1995) has demonstrated the difficultly of negation for non-neural methods like TF-IDF (Sparck Jones, 1972) and BM25 (Robertson et al., 1995) when used out of the box.

To the best of our knowledge, there is little to no published work on negation for neural models. The most similar area in IR is that of argument retrieval (Wachsmuth et al., 2018; Bondarenko et al., 2022), also included in the BEIR dataset, whose aim is to find a counterargument for the given query. However, these datasets implicitly ask the model to find the counterargument to the query through the task design and specifically don't include negation in the query. So although argument retrieval datasets contain a larger amount of negations compared to standard IR datasets like MSMarco (Nguyen et al., 2016), negation is not a conscious choice in the design of either the documents or the queries and is confounded by the implicit task definition. In contrast, we explicitly provide and measure the impact of negation on both documents and queries.

Another recent work by Opitz and Frank (2022) incorporates features from Abstract Meaning Representation (AMR) parsing (including negation, as one of many) to improve SBERT training. However, they only evaluate negation for AMR parsing (and on AMR datasets) whereas we focus on negation in IR and create a benchmark for ranking.

2.5 Contrastive Evaluation

Contrastive evaluation has emerged as a promising evaluation technique: constructing datasets that consist of minor differences but that test crucial distinctions (Gardner et al., 2020; Kaushik et al., 2019). For IR specifically, this has included testing sentence order (Rau and Kamps, 2022), lexical structures (Nikolaev and Padó, 2023), general axiom creation (Völske et al., 2021), paraphrases, mispellings, and ordering (Penha et al., 2022), LLM-based query and document expansion (Weller et al., 2023a), and factuality, formality, fluency, etc. (MacAvaney et al., 2022). We follow these works by evaluating not on a classical IR evaluation corpus, but rather with paired queries and documents.

3 Creating NevIR

We test negation in neural IR using a contrastive evaluation framework, which has shown great utility in understanding neural models (Section 2.5).

3.1 Contrastive Documents

We start by collecting pairs of documents that differ as minimally as possible but include negation, using the CondaQA (Ravichander et al., 2022) dataset as a starting point. CondaQA consists of "in-the-wild" natural paragraphs that contain negation and human-edited versions of those paragraphs that either paraphrase, change the scope of the negation, or undo the negation. For our IR benchmark, we exclude the paraphrase edits, as they do not provide different semantic meanings for comparison. Thus, this allows us to compare the effect of the negation between document pairs with a minimal lexical

Statistic	Train	Dev	Test
# Pairs	948	225	1383
Question 1 Length	10.9	11.1	11.0
Question 2 Length	11.2	11.4	11.4
Average Length Diff	0.95	1.05	1.01
Document 1 Length	112.5	113.0	113.7
Document 2 Length	115.6	116.8	116.8
Average Length Diff	4.39	4.71	4.16

Table 1: NevIR statistics, where length is measured in words. Note that the average length differences only take into account total length; for the distribution of unique word differences see Figure 3.

difference (see Table 1 and Figure 3 for statistics).

3.2 Collecting Contrastive Queries

To test whether IR models correctly rank the documents, we collect natural language queries for those document using workers on Amazon's Mechanical Turk. We ask workers to create one query for each of the two paragraphs, with four constraints:

1. The answer to the queries are the same for both paragraphs
2. The question is answered by a span (e.g. not a yes/no or boolean answer)
3. The question contains enough information to identify the relevant passage from a collection of documents (e.g. it contains relevant entity names, not just "when was he born?")
4. The question can **only be answered by one** of the two paragraphs (thus making the other paragraph irrelevant)

Note that boolean questions would be relevant to both documents, and hence they were excluded. To help annotators understand the task, we allowed them to test their queries against a small neural cross-encoder model (*all-mpnet-base-v2* from Reimers and Gurevych (2019)) but did not require them to. The annotation interface is in Appendix A.

Through a series of initial pilot HITs, we found that annotators would typically quote verbatim from the passage and use the words that were only present in only one document. To prevent models from exploiting this shallow heuristic, we included a 5th constraint: not allowing workers to use any word in the query that was only present in one of the two documents. Note that this was an effective but not perfect constraint (as is shown by TF-IDF's 2% performance in Table 2), as any non-exact string match including subwords, plural versions, etc. would pass this validation check.

We recruited annotators with greater than 99% HIT acceptance rate and greater than 5000 completed HITs. All annotators participated in two paid trial HITs where their work was assessed before moving on. Workers were paid $2.5 USD for approximately six minutes per HIT, for an average of $15 USD per hour. Overall, we had 28 unique annotators with an average of 91 query pairs each.

3.3 Dataset Statistics

Dataset statistics are in Table 1, showing that the average number of words is around 11 for questions and 113 for documents. The average difference in word length between questions and documents is 1 and 4 respectively, showing that items in each pair are nearly the same length. The distribution of unique word differences between queries and documents is in Figure 3 and shows that most queries have small differences of 2 to 5 words, although some differ only by a single negation word and some differ by more than five. The difference between the two documents is much more variable, with about 5-10 different words between them.

3.4 Human Performance

To verify that this dataset is trivial for humans, we asked three annotators to perform the ranking task on 10 randomly sampled test instances. In all three cases, all human annotators ranked all queries correctly, indicating the simplicity of the task.

4 Experimental Settings

4.1 Metric

In early investigations we observed that IR models tended to rank one document above the other for both queries. This motivates our usage of a *pairwise accuracy* score to avoid score inflation when models don't actually understand the negation. We start by having the IR model rank both documents for each query. Then, if the model has correctly ranked the documents for both queries (flipping the order of the ranking when given the negated query) we know that the model has correctly understood the negation and the pair is marked as correct.

4.2 Models

We evaluate a wide variety of models in order to show a comprehensive evaluation across common

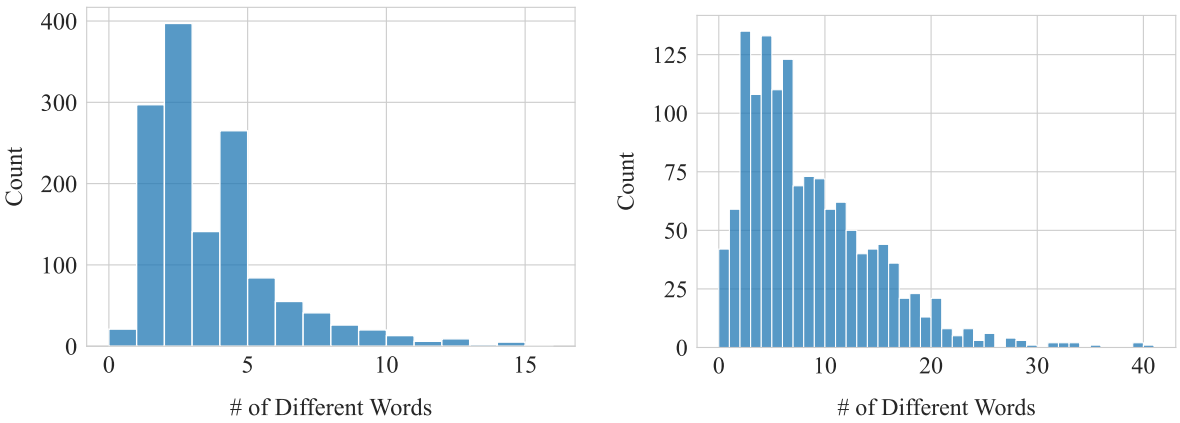

Figure 3: The distribution of the number of different (e.g. unique) words between the queries (left) or documents (right) in each pair. The average length differences are shown in Table 1.

neural IR model types. We note that although there are other models we do not use (as well as many different strategies for model training), all the major types of retrieval models are accounted for here. We evaluate on the following IR model categories:

Sparse We evaluate sparse IR models that use the bag-of-words representation during retrieval. This includes TF-IDF (the only non-neural IR method, here as a baseline), and two variants of SPLADE v2++ (Formal et al., 2022, 2021; Lassance and Clinchant, 2022), the ensemble distillation and self-distillation methods. Note that other variants of SPLADE perform worse than these two methods. We do not include BM25 as implementations of BM25 perform similar to TF-IDF due to the small collection and lexical similarity within the pair.

Late Interaction Late interaction models like ColBERT (Khattab and Zaharia, 2020; Santhanam et al., 2022b) embed documents and queries into one vector for each sub-word token. At inference time, these models need to compute a MaxSim operation between query vectors and document vectors to determine similarity. We use both ColBERT v1 and v2 in our experiments.[3]

Bi-Encoders Another common category of IR models are bi-encoders, which embed both documents and queries into a single vector representation. At inference time the similarity is computed via a simple dot product or cosine similarity. Due to the popularity of this category, we include a broad spectrum: models from SentenceTransformer (Reimers and Gurevych, 2019) trained on MSMarco and/or Natural Questions, DPR (Karpukhin et al., 2020), CoCondenser (Gao and Callan, 2022), and RocketQA (Qu et al., 2021; Ren et al., 2021). Note that these models span a wide variety of pre-training tasks, base models, and complex training/additional fine-tuning strategies like hard negative mining and distillation.

Cross-Encoders Cross-encoders encode both the document and query at the same time, computing attention across both pieces of text. This type of representation is the most expressive but also the most time-intensive, especially for larger models. We use various SentenceTransformer cross-encoders including those trained on MSMarco and various NLI datasets (Demszky et al., 2018; Williams et al., 2018; Cer et al., 2017), RocketQAv2 cross-encoders (Qu et al., 2021; Ren et al., 2021), as well as MonoT5 cross-encoders (Nogueira et al., 2020). Note that MonoT5 models are significantly larger (up to 33x larger for 3B) and more expensive than the other cross-encoders.[4]

Random We include a baseline that randomly ranks the two documents. Since there are two pairs, the expected mean pairwise accuracy is 25% ($\frac{1}{2} * \frac{1}{2}$).

5 Results

5.1 Main Results

The main results are presented in Table 2. We see that the more expressive the representation, the better the models generally perform.

[3]We reproduce ColBERT v1 weights from their repository. We do not use PLAID (Santhanam et al., 2022a) or quantization as there are only two documents in the collection per query and thus no efficiency requirements.

[4]T5 models are also typically used for generative retrieval (GR) (Tay et al., 2022); thus we do not evaluate GR methods since (1) T5 is evaluated with MonoT5 already and (2) GR has been shown to be unable to scale to standard-sized collections (Pradeep et al., 2023) and is not used in practice.

Type	Data	Params	Model Name	Score
Random	N/A	0	Random	25%
Sparse	N/A	N/A	TF-IDF (Pedregosa et al., 2011)	2.0%
	MSMarco	110M	SPLADEv2 ensemble-distill (Formal et al., 2022)	8.0%
	MSMarco	110M	SPLADEv2 self-distill (Formal et al., 2022)	8.7%
Late Interaction	MSMarco	110M	ColBERTv2 (Santhanam et al., 2022b)	13.0%
	MSMarco	110M	ColBERTv1 (Khattab and Zaharia, 2020)	19.7%
Bi-Encoders	NQ	219M	DPR (Karpukhin et al., 2020)	6.8%
	MSMarco	110M	msmarco-bert-base-dot-v5	6.9%
	MSMarco	110M	coCondenser (Gao and Callan, 2022)	7.7%
	MSMarco	85M	RocketQA v2 (Ren et al., 2021)	7.8%
	NQ	66M	nq-distilbert-base-v1	8.0%
	MSMarco	110M	all-mpnet-base-v2	8.1%
	MSMarco	66M	msmarco-distilbert-cos-v5	8.7%
	MSMarco	170M	RocketQA v1 (Qu et al., 2021)	9.1%
	QA Data	110M	multi-qa-mpnet-base-dot-v1	11.1%
Cross-Encoders	MSMarco	85M	RocketQA v2 (Ren et al., 2021)	22.4%
	STSB	355M	stsb-roberta-large	24.9%
	MSMarco	303M	RocketQA v1 (Qu et al., 2021)	26.3%
	MSMarco	61M	MonoT5 small (Nogueira et al., 2020)	27.7%
	MNLI	184M	nli-deberta-v3-base	30.2%
	QNLI	110M	qnli-electra-base	34.1%
	MSMarco	223M	MonoT5 base (default) (Nogueira et al., 2020)	34.9%
	MSMarco	737M	MonoT5 large (Nogueira et al., 2020)	45.8%
	MSMarco	2.85B	MonoT5 3B (Nogueira et al., 2020)	50.6%

Table 2: Results for pairwise contrastive evaluation using paired accuracy. All models are from sentence-transformers (Reimers and Gurevych, 2019) unless otherwise cited. Data indicates the main source of training data for the model, while score indicates Pairwise Accuracy (see Sec 4.1). Note that RocketQA includes both a cross-encoder and bi-encoder for both versions. TF-IDF scores were designed to be low in the task instruction (Section 3.2).

No bi-encoder architecture scores higher than 12% paired accuracy despite the method of pre-training (e.g. CoCondenser) or the type of contrastive training data (MSMarco, NQ, etc.) with most models performing in the 5-10% range.

In the sparse category, we see that TF-IDF scored only 2% paired accuracy. Since we did not allow annotators to use words that were in only one of the paragraphs, this is to be expected.[5] For neural sparse models, all SPLADEv2++ models perform similarly to the bi-encoders, at around 8% paired accuracy.

The late interaction style models perform significantly better than bi-encoders and sparse models, with ColBERTv1 scoring 19.7% and ColBERTv2 scoring 13.0%. Due to the nature of this model we are able to visualize the MaxSim operator to understand its performance (Section 5.3).

The cross-encoder models performed the best, with MonoT5 (the default “base" version) performing at 34.9% paired accuracy (and the largest version at 50.6%). Interestingly, the cross-encoders trained on NLI datasets generally performed better than cross-encoders trained on MSMarco, likely due to the fact that MSMarco contains little negation while NLI datasets typically do have negation.

Overall, despite the strong scores of these models on various standard IR benchmarks, nearly all models perform worse than randomly ranking. Only a handful of cross-encoder models perform better, and they are the slowest and most expensive category of retrieval models. Even these models however, perform significantly below humans and have far from ideal performance.

[5]Note that the 2% performance, instead of 0%, is due to our annotation interface not restricting partial matches (e.g. ‘version" vs “versions", “part" vs “parting" etc.).

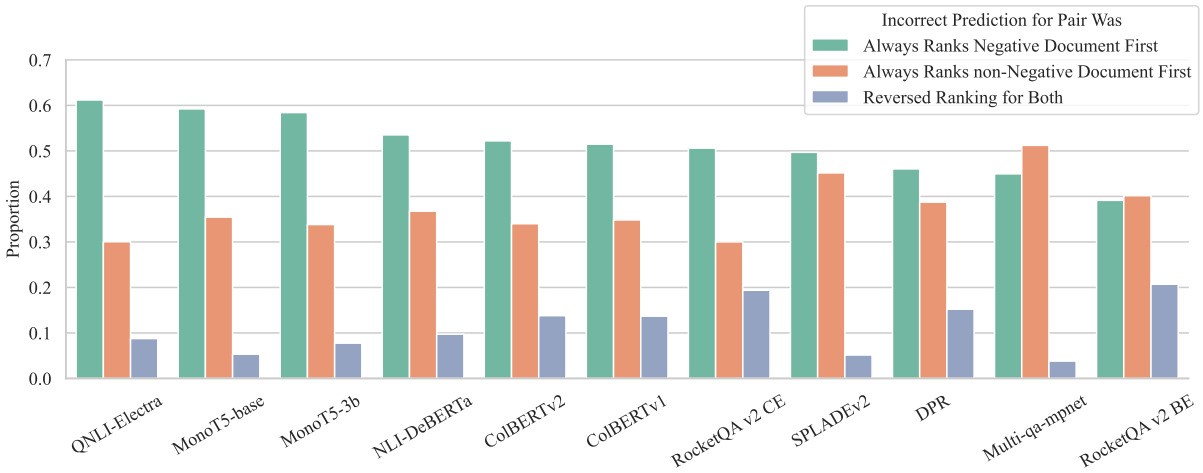

Figure 4: Error analysis of the model predictions, detailing whether models preferred (e.g. by ranking first for both queries) the document with negation (green), the edited non-negation document (orange), or predicted the reversed ranking for both queries (blue). Models that performed better generally preferred negation documents when they made incorrect predictions while bi-encoder models were more balanced in their errors.

5.2 How does model size affect the results?

We note that Table 2 includes different sizes of MonoT5. We see that as model size increases, so does the accuracy (from around 28% with MonoT5-small to around 51% for MonoT5-3B). This aligns with results shown in the natural language processing community about model size (McKenzie et al., 2022; Wei et al., 2022; Ravichander et al., 2022; Weller et al., 2023b).

However, unlike NLP, IR is typically more latency constrained. Thus, models like MonoT5-3B are only feasible for re-ranking and not for first-stage retrieval (c.f. Section 7 for more discussion).

5.3 ColBERT analysis

As ColBERT models provide token-level vectors and use the MaxSim operator, we are able to visualize whether the max operator pays attention to the negation words (Figures 9 and 10 in the appendix, due to space constraints). We find in all sampled instances that the MaxSim operator in ColBERTv1 ignores negation words, not selecting them as the max for any query token. Thus, with default training this is a crucial flaw when it comes to processing negation, which causes its less-than-random performance. However, it is possible to fine-tune these representations to put more weight on the negation words so that the MaxSim correctly identifies them, as seen in Section 6.

5.4 Error Analysis

We conduct an error analysis to determine which document models prefer for a given pair. Models can prefer (e.g. rank highest in both queries) the document with negation, the edited non-negation document, or predict the reversed rank for both queries. Figure 4 shows that the models trained on NLI (and cross-encoders) greatly preferred the document with negation, while bi-encoder models tended to prefer them equally. Reversed rankings are uncommon, with bi-encoder models having the highest percentage (e.g. RocketQA at ∼20%).

6 Fine-Tuning on NevIR

Table 2 shows that models trained on standard IR training datasets do not show strong results on NevIR. However, none of the standard IR datasets include much negation in their queries (potentially due to production systems biasing users, c.f. Section 7). Thus, in this section we fine-tune IR models on NevIR's training set to see how negation-specific training data improves performance.

We use the top performing model from non-sparse categories: multi-qa-mpnet-base-dot-v1 from SentenceTransformers, ColBERTv1, and MonoT5-base from PyGaggle. We fine-tune them using SentenceTransformers, the original ColBERTv1 code, and the original PyGaggle code. We train for 20 epochs and evaluate them on NevIR test and MSMarco dev after each epoch.

Figure 5 shows that fine-tuning on negation data improves performance significantly, but still leaves a large gap to perfect (and the human score of) 100% paired accuracy. As would be expected, the large MonoT5 model quickly learns and then overfits to the data (while quickly losing perfor-

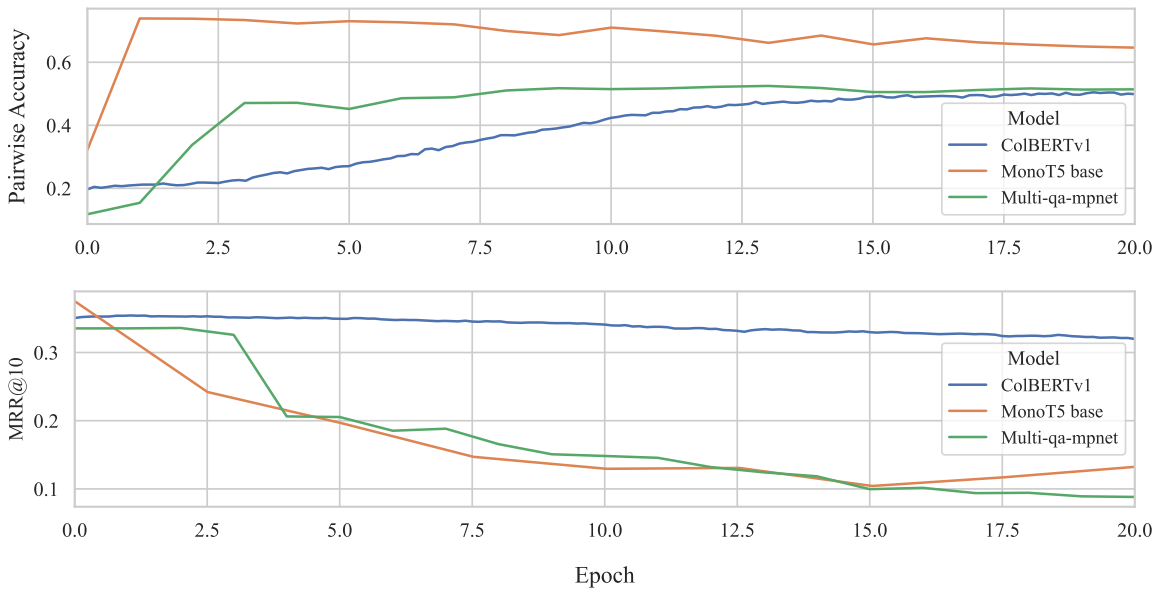

Figure 5: How fine-tuning on NevIR's training set affects results on NevIR and MSMarco: upper shows NevIR's pairwise accuracy scores on test while training for up to 20 epochs, lower shows MSMarco dev MRR@10 scores. For QNLI-electra-base see Appendix E.

mance on MSMarco). Interestingly, ColBERT takes much longer to learn (due to the MaxSim operator), slowly increasing over nearly 20 epochs to learn what the bi-encoder model quickly learned in less than 3. However, we find that ColBERT has a much lower and slower drop in ranking scores on MSMarco (Figure 5 lower). We show visualizations of the MaxSim operator before and after NevIR training in Appendix D, illustrating that before training the MaxSim operator ignores negation, while after training it learns to correctly include it.

7 Discussion and Implications

Implication for Current Systems IR model's performance on NevIR indicates that first stage retrievers do not take negation into account when doing retrieval. Thus, to perform well on negation with current models, expensive cross-encoder rerankers are necessary but not sufficient to achieve good results. Furthermore, our analysis indicates that in order to best learn negation (and significantly improve their performance), models should incorporate negation into their training data.

Thus, when high precision for negation retrieval is *not needed* (e.g. some first stage retrieval settings), current models may be effective, as they will retrieve lexically similar documents regardless of negation. However, in order to have *high-precision* retrieval with negation (and documents with both negation and non-negation have high lexical overlap), expensive cross-encoders are the only current models that perform better than random ranking. NevIR provides the only dataset for measuring and improving retrieval with negation.

Implications for Current Users Anecdotally, most users tend to avoid using negation queries in production IR systems like Google Search. This may be a self-reinforcing problem, as users have found poor results when they use negation in search and hence avoid using negations in the future. For example, the webpage for the University of Utah article that is shown in Figure 1 has since been updated and currently includes no negation words.

Thus, it is unclear whether queries with negation are less common because of people's actual information needs or because production systems have biased users (and content creators) into an avoidance of negation. We hope that by introducing a benchmark for IR evaluation we can help enable these types of queries in the future.

8 Conclusion

We proposed to benchmark negation in neural information retrieval and built a benchmark called NevIR to explore this problem, crowdsourcing annotations from Mechanical Turk. We found that modern IR models perform poorly on this task, with cross-encoder models performing the best (slightly above random performance) and all other architectures (bi-encoder, sparse, and late-

interaction) performing worse than random. Further we showed that simply including negation in fine-tuning provides significant gains, although there is still room for improvement to reach human performance. We hope that this benchmark inspires future work into improving information retrieval model's ability to recognize negation.

9 Limitations

Our work provides results for a broad range of IR models (including the most common and popular), but does not provide results for all possible IR models due to space and time. We welcome future research into investigating alternative methods and models to improve performance on NevIR.

Our dataset follows previous work in designing contrastive evaluation datasets (Kaushik et al., 2019; Penha et al., 2022; MacAvaney et al., 2022) and we note that because of this our work does not provide a large-scale collection to go along with our queries (enabling an analysis of recall along with the precision we measure), as might be found in classic IR datasets. However, as shown by a large body of work (see Section 2.5), contrastive evaluations can provide important insight into understanding and improving neural models. We leave large collection creation with negation and analysis of recall performance to future work.

Acknowledgements

OW is supported by the National Science Foundation Graduate Research Fellowship Program.

References

Mordechai Averbuch, Tom H Karson, Benjamin Ben-Ami, Oded Maimon, and Lior Rokach. 2004. Context-sensitive medical information retrieval. In *MEDINFO 2004*, pages 282–286. IOS Press.

Alexander Bondarenko, Maik Fröbe, Johannes Kiesel, Shahbaz Syed, Timon Gurcke, Meriem Beloucif, Alexander Panchenko, Chris Biemann, Benno Stein, Henning Wachsmuth, et al. 2022. Overview of touché 2022: argument retrieval. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 13th International Conference of the CLEF Association, CLEF 2022, Bologna, Italy, September 5–8, 2022, Proceedings*, pages 311–336. Springer.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Dorottya Demszky, Kelvin Guu, and Percy Liang. 2018. Transforming question answering datasets into natural language inference datasets. *ArXiv preprint*, abs/1809.02922.

Allyson Ettinger. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2022. From distillation to hard negative sampling: Making sparse neural ir models more effective. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 2353–2359, New York, NY, USA. Association for Computing Machinery.

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2021. Splade v2: Sparse lexical and expansion model for information retrieval.

Luyu Gao and Jamie Callan. 2022. Unsupervised corpus aware language model pre-training for dense passage retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2843–2853, Dublin, Ireland. Association for Computational Linguistics.

Matt Gardner, Yoav Artzi, Victoria Basmova, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, et al. 2020. Evaluating models' local decision boundaries via contrast sets. *arXiv preprint arXiv:2004.02709*.

Mareike Hartmann, Miryam de Lhoneux, Daniel Hershcovich, Yova Kementchedjhieva, Lukas Nielsen, Chen Qiu, and Anders Søgaard. 2021. A multilingual benchmark for probing negation-awareness with minimal pairs. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 244–257, Online. Association for Computational Linguistics.

Hangfeng He, Federico Fancellu, and Bonnie Webber. 2017. Neural networks for negation cue detection in Chinese. In *Proceedings of the Workshop Computational Semantics Beyond Events and Roles*, pages 59–63, Valencia, Spain. Association for Computational Linguistics.

Md Mosharaf Hossain, Dhivya Chinnappa, and Eduardo Blanco. 2022. An analysis of negation in natural language understanding corpora. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 716–723, Dublin, Ireland. Association for Computational Linguistics.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. 2019. Learning the difference that makes a difference with counterfactually-augmented data. *arXiv preprint arXiv:1909.12434*.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 39–48. ACM.

Mi-Young Kim, Juliano Rabelo, and Randy Goebel. 2019. Statute law information retrieval and entailment. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*, pages 283–289.

Young-Whan Kim and Jin H. Kim. 1990. A model of knowledge based information retrieval with hierarchical concept graph. *J. Documentation*, 46:113–136.

Carlos Lassance and Stéphane Clinchant. 2022. An efficiency study for splade models. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 2220–2226, New York, NY, USA. Association for Computing Machinery.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Shoushan Li and Chu-Ren Huang. 2009. Sentiment classification considering negation and contrast transition. In *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation, Volume 1*, pages 307–316, Hong Kong. City University of Hong Kong.

Sean MacAvaney, Sergey Feldman, Nazli Goharian, Doug Downey, and Arman Cohan. 2022. ABNIRML: Analyzing the behavior of neural IR models. *Transactions of the Association for Computational Linguistics*, 10:224–239.

Ian McKenzie, Alexander Lyzhov, Alicia Parrish, Ameya Prabhu, Aaron Mueller, Najoung Kim, Sam Bowman, and Ethan Perez. 2022. Inverse scaling prize: First round winners.

April R McQuire and Caroline M Eastman. 1998. The ambiguity of negation in natural language queries to information retrieval systems. *Journal of the American Society for Information Science*, 49(8):686–692.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. *choice*, 2640:660.

Dmitry Nikolaev and Sebastian Padó. 2023. Representation biases in sentence transformers. *arXiv preprint arXiv:2301.13039*.

Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document ranking with a pre-trained sequence-to-sequence model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718, Online. Association for Computational Linguistics.

Juri Opitz and Anette Frank. 2022. SBERT studies meaning representations: Decomposing sentence embeddings into explainable semantic features. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 625–638, Online only. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Gustavo Penha, Arthur Câmara, and Claudia Hauff. 2022. Evaluating the robustness of retrieval pipelines with query variation generators. In *European conference on information retrieval*, pages 397–412. Springer.

Ronak Pradeep, Kai Hui, Jai Gupta, Adam D Lelkes, Honglei Zhuang, Jimmy Lin, Donald Metzler, and Vinh Q Tran. 2023. How does generative retrieval scale to millions of passages? *arXiv preprint arXiv:2305.11841*.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. In *Proceedings of the 2021 Conference of the North American Chapter of*

the Association for Computational Linguistics: Human Language Technologies, pages 5835–5847, Online. Association for Computational Linguistics.

David Rau and Jaap Kamps. 2022. The role of complex nlp in transformers for text ranking. In *Proceedings of the 2022 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 153–160.

Abhilasha Ravichander, Matt Gardner, and Ana Marasovic. 2022. CONDAQA: A contrastive reading comprehension dataset for reasoning about negation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8729–8755, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, QiaoQiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021. RocketQAv2: A joint training method for dense passage retrieval and passage re-ranking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2825–2835, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at trec-3. *Nist Special Publication Sp*, 109:109.

Keshav Santhanam, Omar Khattab, Christopher Potts, and Matei Zaharia. 2022a. Plaid: an efficient engine for late interaction retrieval. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 1747–1756.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022b. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734, Seattle, United States. Association for Computational Linguistics.

Kurt Shuster, Mojtaba Komeili, Leonard Adolphs, Stephen Roller, Arthur Szlam, and Jason Weston. 2022. Language models that seek for knowledge: Modular search & generation for dialogue and prompt completion. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 373–393, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Tomek Strzalkowski, Jose Perez Carballo, and Mihnea Marinescu. 1995. Natural language information retrieval: Trec-3 report. *NIST SPECIAL PUBLICATION SP*, pages 39–39.

Yi Tay, Vinh Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, et al. 2022. Transformer memory as a differentiable search index. *Advances in Neural Information Processing Systems*, 35:21831–21843.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *ArXiv preprint*, abs/2104.08663.

Michael Völske, Alexander Bondarenko, Maik Fröbe, Benno Stein, Jaspreet Singh, Matthias Hagen, and Avishek Anand. 2021. Towards axiomatic explanations for neural ranking models. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 13–22.

Henning Wachsmuth, Shahbaz Syed, and Benno Stein. 2018. Retrieval of the best counterargument without prior topic knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 241–251, Melbourne, Australia. Association for Computational Linguistics.

Jason Wei, Yi Tay, and Quoc V Le. 2022. Inverse scaling can become u-shaped. *ArXiv preprint*, abs/2211.02011.

Orion Weller, Kyle Lo, David Wadden, Dawn Lawrie, Benjamin Van Durme, Arman Cohan, and Luca Soldaini. 2023a. When do generative query and document expansions fail? a comprehensive study across methods, retrievers, and datasets. *arXiv preprint arXiv:2309.08541*.

Orion Weller, Marc Marone, Nathaniel Weir, Dawn Lawrie, Daniel Khashabi, and Benjamin Van Durme. 2023b. " according to..." prompting language models improves quoting from pre-training data. *arXiv preprint arXiv:2305.13252*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

A Annotation Interface

In Figure 6 we show the annotation interface provided to workers on Mechanical Turk.

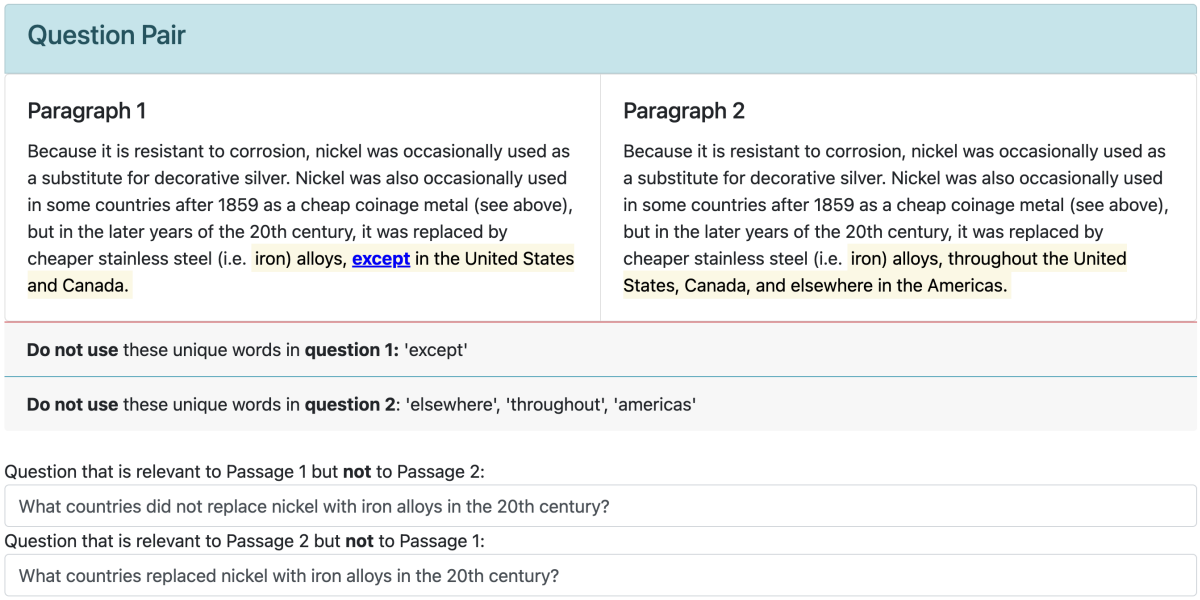

Figure 6: Number of unique words between the two queries.

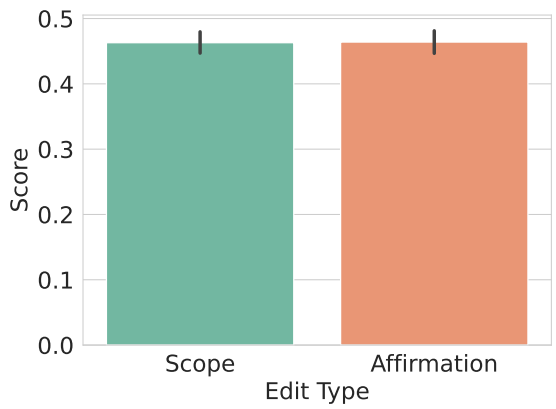

Figure 7: Edit types from the CondaQA dataset and their average pairwise scores. Error bars indicate a 95% confidence interval.

B Document Edit Types

We also analyze the edit types from the original CondaQA dataset to see if they impact the pairwise accuracy. We see in Figure 7 that there is no statistical difference (given the 95% confidence interval) between the two types of edits for the MonoT5-3B model (and we note that other models are similar and hence we only include one model).

C Cosine Similarity after Fine-Tuning

In Figure 8 we see the results for cosine similarity between each document pair during different epochs. We can see that the representations start nearly identically, but shift to be further apart and to have more variance as training continues. This plot was created using the multi-qa-mpnet model, but other dense models show similar results.

D ColBERT Analysis

We show two heatmaps for ColBERTv1 models, the first using the original model trained on MS-Marco and the 2nd after fine-tuning for 20 epochs on NevIR. We see in Figure 9 that the model fails to associate any maximum tokens with the crucial word “rather" instead associating “not" with “usually". In contrast, after training on NevIR, the model correctly associates “rather" with “not".

E Results with training QNLI-electra-base on NevIR

Figure 11 shows results with QNLI-electra-base also, which shows similar results to MonoT5 in the main paper. We do not show results for MSMarco as QNLI-electra-base was not trained on MSMarco.

F Importance of Negation in Retrieval

We include pictures of the tweet referenced at https://x.com/soft/status/1449406390976409600 in Figure 12, showing the dangers of not understanding negation.

G Hyperparameters and Computational Resources

All experiments were run on a cluster of V100s with each experiment taking less than an hour on

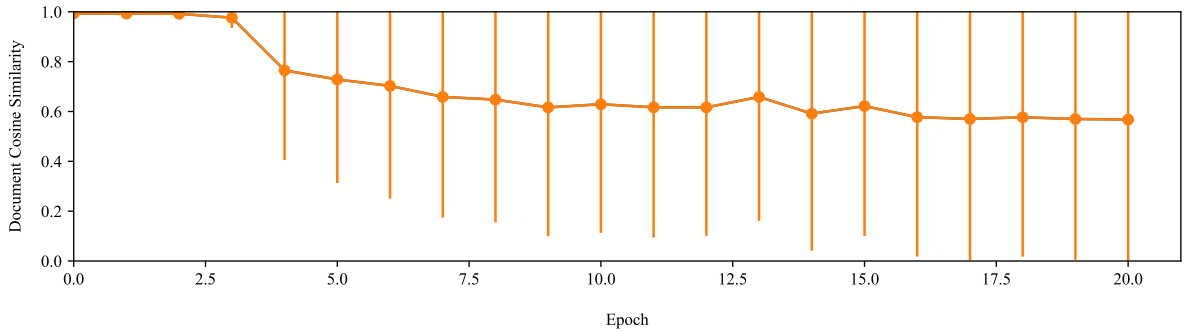

Figure 8: Cosine similarity scores between documents in the pairs during fine-tuning for the multi-qa-mpnet bi-encoder model. Error bars indicate one standard deviation.

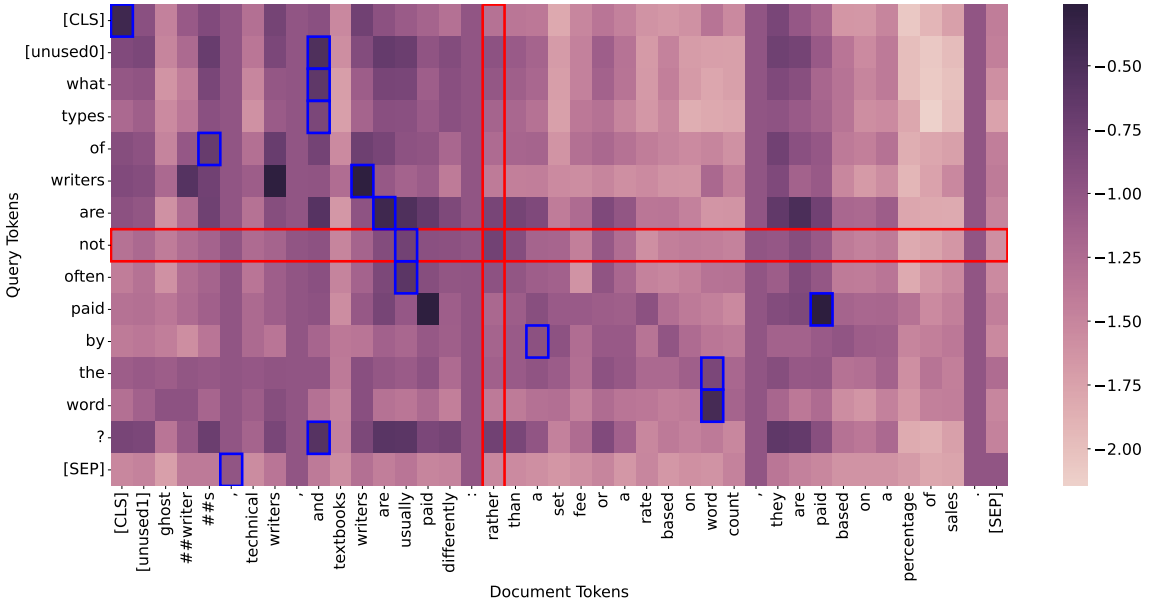

Figure 9: An example instance with results from ColBERT's MaxSim operator from the ColBERTv1 model. Red highlights indicate the tokens corresponding to the negation (or lack of negation) while blue highlights indicate the max token for the MaxSim operator. Note that this model predicts the MaxSim token of "usually" for "not" and has no Max for the crucial word "rather". However, further fine-tuning helps improve this, see Figure 10.

one V100.

We use default hyperparameters for all models for inference (and many models do not have any hyperparameters). For ColBERT training we use their code that has a default learning rate of 3e-6 and for bi-encoder training we use Sentence-Transformers that has a default of 2e-5.

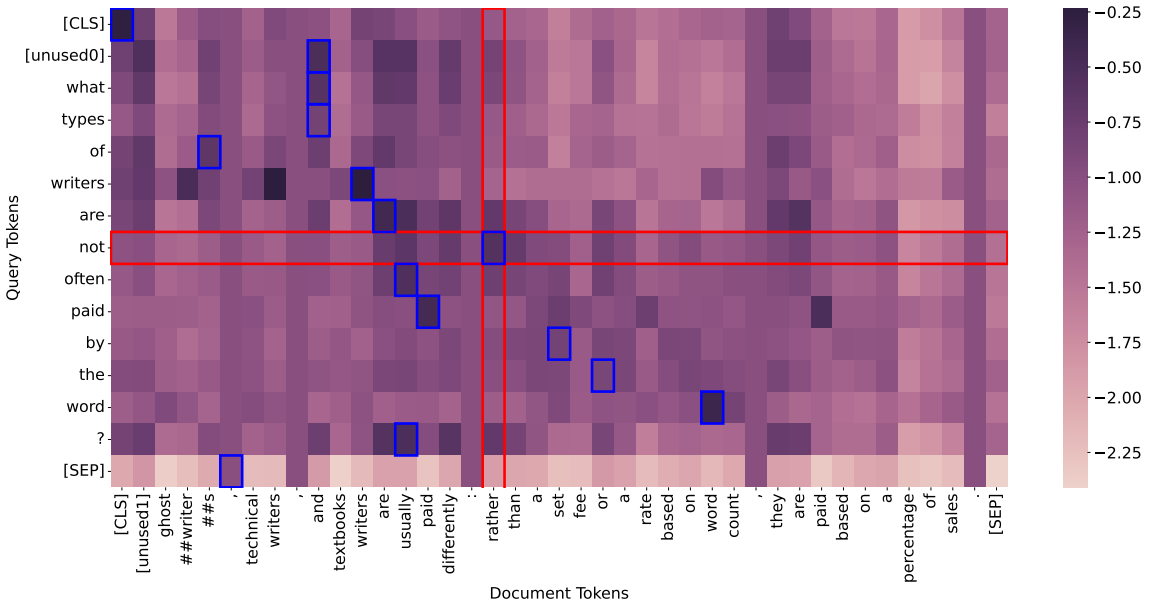

Figure 10: An example instance with results from ColBERT's MaxSim operator from the ColBERTv1 model trained for 20 epochs on NevIR. Red highlights indicate the tokens corresponding to the negation (or lack of negation) while blue highlights indicate the max token for the MaxSim operator. Note that this model correctly associates the word "not" with the crucial word "rather" unlike Figure 9.

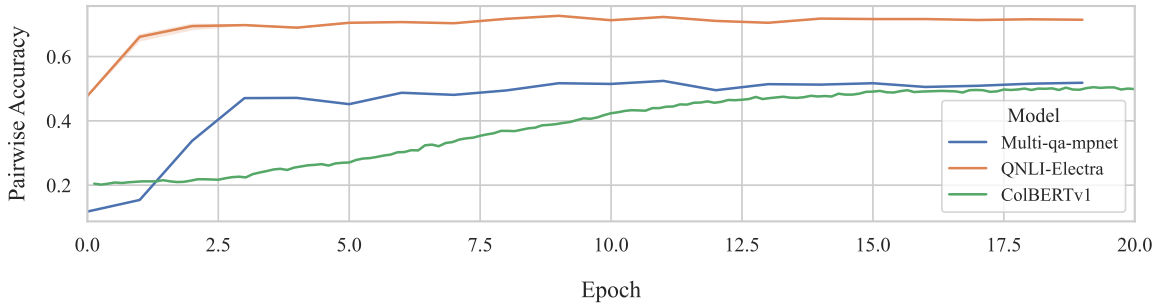

Figure 11: Results from fine-tuning IR models on the NevIR training set, including QNLI-electra-base. The plot shows NevIR test set pairwise accuracy scores while training for up to 20 epochs

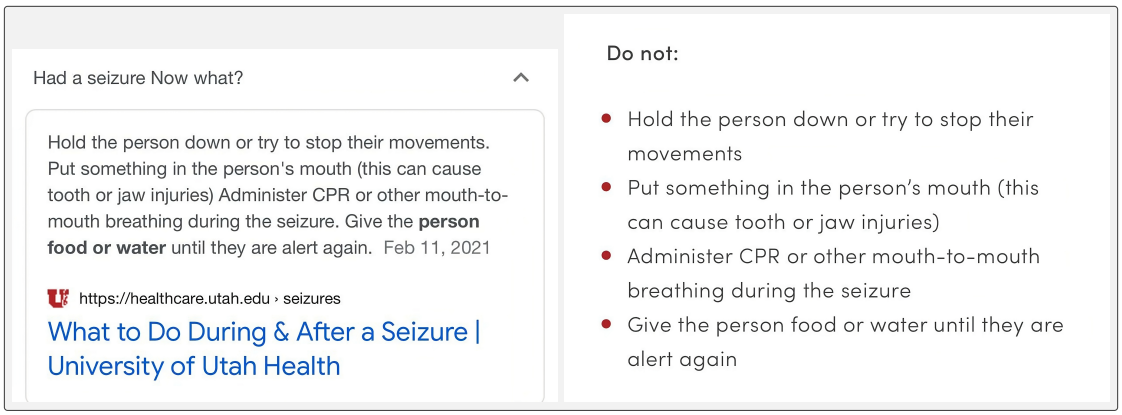

Figure 12: Reproduction of the tweet showing Google Search making a life-threatening recommendation and failing to catch the negation in the article.