ABNIRML: Analyzing the Behavior of Neural IR Models

Sean MacAvaney[†*] **Sergey Feldman**[‡] **Nazli Goharian**[†]
Doug Downey[‡] **Arman Cohan**[‡§]
[†]IR Lab, Georgetown University, Washington, DC, USA
[‡]Allen Institute for AI, Seattle, WA, USA
[§]Paul G. Allen School of Computer Science, University of Washington, WA, USA
{sean,nazli}@ir.cs.georgetown.edu
{sergey,dougd,armanc}@allenai.org

Abstract

Pretrained contextualized language models such as BERT and T5 have established a new state-of-the-art for ad-hoc search. However, it is not yet well understood why these methods are so effective, what makes some variants more effective than others, and what pitfalls they may have. We present a new comprehensive framework for Analyzing the Behavior of Neural IR ModeLs (ABNIRML), which includes new types of diagnostic probes that allow us to test several characteristics—such as writing styles, factuality, sensitivity to paraphrasing and word order—that are not addressed by previous techniques. To demonstrate the value of the framework, we conduct an extensive empirical study that yields insights into the factors that contribute to the neural model's gains, and identify potential unintended biases the models exhibit. Some of our results confirm conventional wisdom, for example, that recent neural ranking models rely less on exact term overlap with the query, and instead leverage richer linguistic information, evidenced by their higher sensitivity to word and sentence order. Other results are more surprising, such as that some models (e.g., T5 and ColBERT) are biased towards factually correct (rather than simply relevant) texts. Further, some characteristics vary even for the same base language model, and other characteristics can appear due to random variations during model training.[1]

1 Introduction

Pre-trained contextualized language models such as BERT (Devlin et al., 2019) are state-of-the-art for a wide variety of natural language processing tasks (Xia et al., 2020). In Information Retrieval (IR), these models have brought about large improvements in the task of *ad-hoc retrieval*—ranking documents by their relevance to a textual query (Lin et al., 2020; Nogueira and Cho, 2019; MacAvaney et al., 2019a; Dai and Callan, 2019b)—where the models increasingly dominate competition leaderboards (Craswell et al., 2019; Dalton et al., 2019).

Despite this success, little is understood about *why* pretrained language models are effective for ad-hoc ranking. Previous work has shown that traditional IR axioms, for example, that increased term frequency should correspond to higher relevance, do *not* explain the behavior of recent neural models (Câmara and Hauff, 2020). Outside of IR, others have examined what characteristics contextualized language models learn in general (Liu et al., 2019a; Rogers et al., 2020; Loureiro et al., 2020), but it remains unclear if these qualities are valuable for ad-hoc ranking specifically. Thus, new approaches are necessary to characterize models.

We propose a new framework aimed at Analyzing the Behavior of Neural IR ModeLs (ABNIRML[2]), which aims to probe the sensitivity of ranking models on specific textual properties. Probes consist of samples comprising a query and two contrastive documents. We propose three strategies for building probes. The ''measure and match'' strategy (akin to the diagnostic datasets proposed by Rennings et al. [2019]) constructs probing samples by controlling one measurement (e.g., term frequency) and varying another (e.g., document length) using samples from an existing IR collection. Unlike Rennings et al. (2019), our framework generalizes the idea to any measurable characteristic, rather than relying chiefly on prior proposed IR axioms. A second strategy, ''textual

[*]Currently at the University of Glasgow. Work done in part during an internship at the Allen Institute for AI.

[1]Code: https://github.com/allenai/abnriml.

[2]Pronounced /ab'nərməl/ , similar to ''abnormal''.

Transactions of the Association for Computational Linguistics, vol. 10, pp. 224–239, 2022. https://doi.org/10.1162/tacl_a_00457
Action Editor: Tim Baldwin. Submission batch: 7/2021; Revision batch: 10/2021; Published 3/2022.

manipulation,'' probes the effect that altering the text of a document text has on its ranking. Finally, a ''dataset transfer'' strategy constructs probes from non-IR datasets. The new probes allow us to isolate model characteristics—such as sensitivity to word order, degree of lexical simplicity, or even factuality—that cannot be analyzed using other approaches.

Using our new framework, we perform the first large-scale analysis of neural IR models. We compare today's leading ranking techniques, including those using BERT (Devlin et al., 2019) and T5 (Raffel et al., 2020), methods focused on efficiency like DocT5Query (Nogueira et al., 2020) and EPIC (MacAvaney et al., 2020), and dense retrieval models like ANCE (Xiong et al., 2021) and ColBERT (Khattab and Zaharia, 2020).[3] Some of our results establish widely believed, but not-yet-verified, conjectures about neural models. For example, we show that neural models can exploit richer linguistic signals than classical term-matching metrics like BM25: When controlling for term frequency match, the neural models detect document relevance much more accurately than the BM25 baseline. Similarly, unlike prior approaches, rankers based on BERT and T5 are heavily influenced by word order: Shuffling the words in a document consistently lowers the document's score relative to the unmodified version, and neural rankers show a sensitivity to sentence order that is completely absent in classical models. Other findings from ABNIRML are more surprising. For example, we find that the T5 and ColBERT models we examine prefer answers that are factually correct, implying that they encode and utilize some real-world knowledge. Further, although this knowledge may be a result of the model's pre-training process, it is not necessarily utilized as a ranking signal, given that other models that use the same base language model do not have the same preference. Our battery of probes also uncover a variety of other findings, including that adding additional text to documents can often exhibit adverse behavior in neural models—decreasing the document's score when the added text is relevant, and increasing the score when the added text is irrelevant.

[3]Although a multitude of other models exist, it is impractical to investigate them all. We instead focus on a representative sample of the recent and successful models and well-known baselines to provide context.

In summary, we present a new framework (ABNIRML) for performing analysis of ad-hoc ranking models. We then demonstrate how the framework can provide insights into ranking model characteristics by providing the most comprehensive analysis of neural ranking models to date. Our software implementation of the framework is easily extensible, facilitating the replication of our results and further analyses in future work.

2 ABNIRML

In order to characterize the behavior of ranking models we construct several diagnostic probes. Each probe aims to evaluate specific properties of ranking models and probe their behavior (if they are heavily influenced by term matching, discourse and coherence, conciseness/verbosity, writing styles, etc.). We formulate three different approaches to construct probes (Measure and Match, Textual Manipulation, and Dataset Transfer).

In ad-hoc ranking, a query (expressed in natural language) is submitted by a user to a search engine, and a ranking function provides the user with a list of natural language documents sorted by relevance to the query. More formally, let $R(q, d) \in \mathbb{R}$ be a ranking function, which maps a given query q and document d (each being a natural-language sequence of terms) to a real-valued ranking score. At query time, documents in a collection D are scored using $R(\cdot)$ for a given query q, and ranked by the scores (conventionally, sorted descending by score). Learning-to-rank models optimize a set of parameters for the task of relevance ranking based on training data.

2.1 Document Pair Probing

We utilize a *document pair probing* strategy, in which probes are composed of samples, each of which consists of a query and two documents that differ primarily in some characteristic of interest (e.g., succinctness). The ranking scores of the two documents are then compared (with respect to the query). This allows the isolation of particular model preferences. For instance, a probe could consist of summarized and full texts of news articles; models that consistently rank summaries over full texts prefer succinct text.

More formally, each document pair probe consists of a collection of samples S, where each $\langle q, d_1, d_2 \rangle \in S$ is a 3-tuple consisting of a query (or query-like text, q), and two documents (or

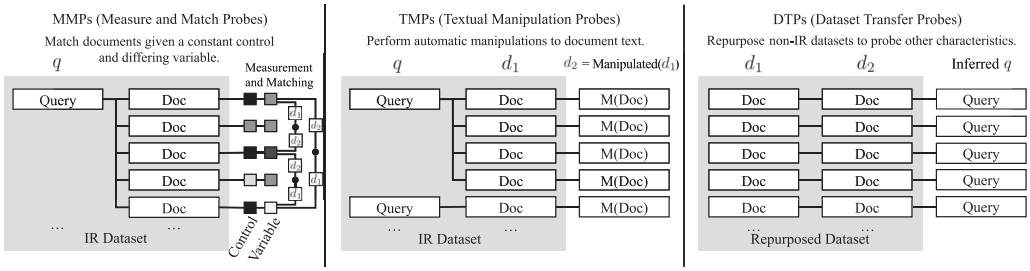

Figure 1: Overview of strategies for constructing probes. Each probe in ABNIRML is composed of samples, each of which consists of a query (q) and two documents (d_1 and d_2).

document-like texts, d_1 and d_2). The relationship between d_1 and d_2 (with respect to q) for each sample defines the probe. For example, a probe testing summarization could be defined as: (1) d_2 is a summary of d_1, and (2) d_1 is relevant to query q.

Almost all of our probes are *directional*, where d_2 has some attribute that d_1 lacks, and we measure the effect of this attribute on ranking. Specifically, each sample in the probe is scored as: $(+1)$ scoring d_1 above d_2 (a *positive* effect), (-1) scoring d_2 above d_1 (a *negative* effect), or (0) a *neutral* effect. Formally, the effect $\mathit{eff}(\cdot)$ of a given sample is defined as:

$$\mathit{eff}(q, d_1, d_2) = \begin{cases} 1 & R(q,d_1) - R(q,d_2) > \delta \\ -1 & R(q,d_1) - R(q,d_2) < -\delta \\ 0 & -\delta \leq R(q,d_1) - R(q,d_2) \leq \delta \end{cases} \quad (1)$$

The parameter δ adjusts how large the score difference between the scores of d_1 and d_2 must be in order to count as positive or negative effect. This allows us to disregard small changes to the score that are unlikely to affect the final ranking. In practice, δ depends on the ranking model because each model scores on different scales. Therefore we tune δ for each model (see Section 3.3).

Symmetric probes are different from directional ones in that d_1 and d_2 are exchangeable; for example, we experiment with one symmetric probe in which d_1 and d_2 are paraphrases of each other. For symmetric probes only the magnitude of score difference is meaningful, and thus eff outputs 1 if the absolute value of the difference is larger than δ, and 0 otherwise.

A model's performance on a particular probe is summarized by a single score s that averages the effect of all samples in the probe:

$$s = \frac{1}{|S|} \sum_{\langle q,d_1,d_2 \rangle \in S} \mathit{eff}(q, d_1, d_2) \quad (2)$$

Note that this score is in the interval $[-1, 1]$ for directional probes and $[0, 1]$ for symmetric probes. For directional probes, positive scores indicate a stronger preference towards documents from group 1 (d_1 documents), and negative scores indicate a preference towards documents from group 2 (d_2 documents). Scores near 0 indicate no strong preference or preferences that are split roughly evenly; disentangling these two cases requires analyzing individual effect scores.

There are several important differences between our setup and the ''diagnostic dataset'' approach proposed by Rennings et al. (2019). First, by including the δ threshold, we ensure that our probes measure differences that can affect the final order in ranked lists. Second, by including the ''neutral effect'' case in our scoring function, we distinguish between cases in which d_1 or d_2 are preferred and cases where neither document is strongly preferred. And finally, our probes are aimed at describing model behavior, rather than evaluating models. For instance, one of our tests measures whether the model prefers succinct or elaborative text—whether this preference is desirable depends on the application or even the particular user.

2.2 Document Pair Probing Strategies

In this work, we explore three strategies for designing document pair probes. As discussed below, the strategies have different strengths and weaknesses. When used in concert, they allow us to characterize a wide variety of model behaviors. Figure 1 provides an overview of the strategies.

2.2.1 Measure and Match Probes (MMPs)

Some surface-level characteristics of documents, such as its Term Frequency (TF) for a given query, are both easy to measure and valuable for characterizing models. Comparing the ranking scores

of two documents that differ in one characteristic but are otherwise similar yields evidence of how the characteristic influences model behavior. Measure and Match Probes (MMPs) follow such an approach. MMPs involve first measuring the characteristics of judged query-document pairs in an IR dataset. Then, the pairs are matched to form probe samples consisting of a *control* (a characteristic that approximately matches between the documents, such as document length), and a *variable* (which differs between documents, such as TF). Probes used in previous work to verify existing ranking axioms (Câmara and Hauff, 2020; Rennings et al., 2019)[4] are instances of MMPs.

For our experiments, we design MMPs to explore the relationship between the primary IR objective (document relevance) and the classical IR ranking signal (TF, potentially controlling for document length). We are motivated to explore this relationship because TF has long been used as a core signal for ranking algorithms; a departure from monotonically increasing the score of a document as TF increases would represent a fundamental shift in the notion of relevance scoring (Fang et al., 2004). Specifically, we explore the following characteristics in MMPs:

- **Relevance**: the human-assessed graded relevance score of a document to the given query.
- **Length**: the document length, in total number of non-stopword tokens.
- **TF**: the individual Porter-stemmed Term Frequencies of non-stopword terms from the query. To determine when the TF of two documents are different, we use the condition that the TF of at least one query term in d_1 must be greater than the same term in d_2, and that no term in d_1 can have a lower TF than the corresponding term in d_2.
- **Overlap**: the proportion of non-stopword terms in the document that appear in the query. Put another way, the total TF divided by the document length.

Each of these characteristics is used as both a variable (matching based on differing values) and a control (matching based on identical values). In our experiments, we examine all pairs of these characteristics, greatly expanding upon IR axioms investigated in prior work.

We note that the MMPs that we explore in this work do not cover all prior IR axioms. For instance, axioms SMTC1–3, proposed by Fang and Zhai (2006), suggest behaviors related to the occurrence of semantically similar terms. Although MMPs can be constructed to test these, we assert that other types of probes are more suitable to testing these behaviors. We test textual fluency, formality, and simplicity (all of which are specific types of semantic similarity) while controlling for the meaning of the text using dataset transfer probes (Section 2.2.3).

2.2.2 Textual Manipulation Probes (TMPs)

Not all characteristics are easily captured with MMPs. For instance, it would be difficult to probe the sensitivity to word order with MMPs; it is unlikely to find naturally occurring document pairs that use the same words but in a different order. Nevertheless, it is valuable to understand the extent to which models are affected by characteristics like this, given that traditional bag-of-words models are unaffected by word order and that there is evidence that word order is unimportant when fine-tuning recent neural models (Sinha et al., 2021; Alleman et al., 2021). To overcome these limitations, we propose Textual Manipulation Probes (TMPs). TMPs apply a manipulation function to scored documents from an existing IR dataset. For example, for probing word order, we can use a simple manipulation function that, given a document d_1, creates a corresponding synthetic document d_2 by shuffling the order of the words in each sentence. The degree to which a model prefers d_1 is then a measure of its preference for proper word order. Prior work that uses a similar approach for probing ranking methods include the collection perturbation tests of Fang et al. (2011) (which perform operations like removing documents from the collection and deleting individual terms from documents) and a diagnostic dataset proposed by Rennings et al. (2019) (which tests the effect of duplicating the document: an adaptation of a traditional ranking axiom). Although TMPs allow probing a wider variety of characteristics than MMPs, we note that they involve constructing artificial data; d_2 may not resemble documents seen in practice. Despite this, their versatility make TMPs an attractive choice for a variety of characteristics.

[4] An example is TFC1 from Fang et al. (2004), which suggests that higher TFs should be mapped to higher scores.

We now detail the specific TMPs we explore in our experiments. We use TMPs to verify a key difference we expect to hold between neural models and previous rankers: Because neural models are pretrained on large bodies of running text, they should make better use of richer linguistic features like word order. We investigate this with TMPs that **shuffle words** in the document. We also probe which aspects of word order are important, through TMPs that only shuffle a small number of non-content words (**prepositions**) and TMPs that only shuffle the **sentence order**, but not the individual words within each sentence. Further, another important distinction of pretrained neural models is that they process unaltered text, without classical normalization like **stopword removal** or **lemmatization**; we introduce TMPs that study these manipulations.[5] Recognizing changes such as lemmatization and word shuffling can drastically alter the text, we also include a more subtle TMP that applies typical typograhpical errors (**typos**) by replacing words with common misspellings.[6] We also evaluate the recent, effective technique of using neural models to add content (**DocT5Query terms** [Nogueira et al., 2020]) to each document to aid IR, and contrast this with a complementary TMP that adds a **non-relevant sentence** to the document.

2.2.3 Dataset Transfer Probes (DTPs)

Even with MMPs and TMPs, some characteristics may still be difficult to measure. For instance, for attributes like textual *fluency* (the degree to which language sounds like a native speaker wrote it), we would need to find pairs of otherwise-similar documents with measurable differences in fluency (for an MMP) or identify ways to automatically manipulate fluency (for a TMP), both of which would be difficult. To probe characteristics like these, we propose Dataset Transfer Probes (DTPs). In this setting, a dataset built for a purpose other than ranking is repurposed to probe a ranking model's behavior. For example, one could create a DTP from a dataset of human-written textual fluency pairs (e.g., from the JFLEG dataset [Napoles et al., 2017]) to sidestep challenges in both measurement and manipulation. Text pair datasets are abundant, allowing us to probe a wide variety of characteristics, like fluency, formality, and succinctness. With these probes, d_1 and d_2 can be easily defined by the source dataset. In some cases, external information can be used to infer a corresponding q, such as using the title of the article as a query for news article summarization tasks, a technique that has been studied before to train ranking models (MacAvaney et al., 2019b). In other cases, queries can be artificially generated, as long as the text resembles a likely query.

We first use DTPs to investigate the important question of whether models exhibit confounding preferences for *stylistic* features of text are at least partially independent of relevance. Specifically, we first investigate paraphrases in general, and then move on to check the specific qualities of fluency, formality, simplicity, lexical bias, and succinctness. We then use DTPs to test the capacity of models to encode and utilize real-world knowledge through probes that measure a model's tendency to select factual answers.

The TMPs described in the previous section probe the sensitivity of models to word order. In this case, the *words* remain the same, but *meaning* is altered. It is natural to wonder whether model behaviors would be similar if the meaning is preserved when using different words. This motivates a **paraphrase** DTP. We construct this probe from the Microsoft Paraphrase Corpus (MSPC).[7] We select d_1 and d_2 from all text pairs labeled as paraphrases. Note that this is the first example of a symmetric probe, as there is no directionality in the paraphrase relation; the assignment of d_1 and d_2 is arbitrary. We generate q by randomly selecting a noun chunk that appears in both versions of the text, ensuring a query that is relevant to both texts. (If no such chunk exists, we discard the sample.) By selecting a noun chunk, the query remains reasonably similar to a real query.

Although the paraphrase probe can tell us whether models distinguish between text with similar meaning, it cannot tell us what characteristics it favors when making such a distinction. To gain insights here, we propose several directional probes based on stylistic differences that result in similar meanings. One such characteristic is textual **fluency**. We propose a DTP using the

[5] We use SpaCy's (Honnibal and Montani, 2017) lemmatizer, rather than, e.g., a stemmer, because the outputs from a stemming function like Porter are often not found in the lexicon of models like BERT.

[6] We use this list of common errors in English text: `https://en.wikipedia.org/wiki/Commonly_misspelled_English_words`.

[7] `https://www.microsoft.com/en-us/download/details.aspx?id=52398`.

JFLEG dataset (Napoles et al., 2017). This dataset contains sentences from English-language fluency tests. Each non-fluent sentence is corrected for fluency by four fluent English speakers to make the text sound ''natural'' (changes include grammar and word usage changes). We treat each fluent text as a d_1 paired with the non-fluent d_2, and use the strategy used for paraphrases to generate q.

We probe **formality** by building a DTP from the GYAFC dataset (Rao and Tetreault, 2018). This dataset selects sentences from Yahoo Answers and has four annotators make edits to the text that either improve the formality (for text that is informal), or reduce the formality (for text that is already formal). We treat formal text as d_1 and informal text as d_2. Because the text came from Yahoo Answers, we can link the text back to the original questions using the Yahoo L6 dataset.[8] We treat the question (title) as q. In cases where we cannot find the original text or there are no overlapping non-stopword lemmas from q in both d_1 and d_2, we discard the sample.

The **simplicity** of text indicates the ease of reading a particular text. We test the effect of *lexical* text simplicity using the WikiTurk dataset provided by Xu et al. (2016). In this dataset, sentences from Wikipedia were edited to make them simpler by Amazon Mechanical Turk workers. We treat the simplified text as d_1, the original text as d_2, and we use the query construction technique from the paraphrase probe for q.

Text can also express similar ideas but with differing degrees of subjectivity or bias. We construct a **neutrality** DTP using the Wikipedia Neutrality Corpus (WNC) dataset (Pryzant et al., 2020). This corpus consists of sentences that were corrected by Wikipedia editors to enforce the platform's neutral point of view. We use the neutral text as d_1, the biased text as d_2, and we use the query construction technique from the paraphrase probe for q.

An idea can also be expressed in greater or lesser detail. To probe whether models have a preference for **succinctness**, we construct DTPs from summarization datasets, using the assumption that a document's summary will be more succinct than its full text. We utilize two datasets to conduct this probe: XSum (Narayan et al., 2018) and CNN/DailyMail (See et al., 2017). The former uses extremely concise summaries from BBC articles, usually consisting of a single sentence. The CNN/DailyMail dataset uses slightly longer bullet point list summaries, usually consisting of around 3 sentences. For these probes, we use the title of the article as q, the summarized text as d_1, and the article body as d_2. When there is no overlap between the non-stopword lemmas of q in both d_1 and d_2, we discard the samples. We further sub-sample the dataset at 10% because the datasets are already rather large. To handle the long full text in BERT and EPIC, we use the passage aggregation strategy proposed by MacAvaney et al. (2019a).

Moving beyond probes that express similar ideas, we explore the extent to which models are aware of real-world knowledge using a **factuality** probe. This probe is motivated by the intuition that contextualized language models may be memorizing facts from the pre-training corpus when determining relevance. We construct this probe from the Natural Questions (Kwiatkowski et al., 2019) dataset. We make use of the known answer text from NQ by replacing it with a similar answer. Similar answers must be of the same entity type[9] and have the same number of non-stopword tokens. We discard samples where the question text contains the answer text (e.g., this-or-that questions). We use the factual text as d_1, the non-factual version as d_2, and the question text as q. Note that this probe can be considered both a DTP and a TMP. We decide to consider it to primarily be a DTP because it makes use of data specific to this external dataset (i.e., answer strings).

3 Experimental Setup

3.1 Datasets

We use the MS-MARCO passage dataset (Campos et al., 2016) to train the neural ranking models. The training subset contains approximately $809k$ natural-language questions from a query log (with an average length of 7.5 terms) and 8.8 million candidate answer passages (with an average length of 73.1 terms). Due to its scale in number of queries, it is shallowly annotated, almost always containing fewer than 3 positive judgments per query. This dataset is frequently

[8] https://webscope.sandbox.yahoo.com/catalog.php?datatype=l&did=11.

[9] Entities extracted using SpaCy: Person (PER), Location (LOC), Geo-Political Entity (GPE), Nationality/Religion/etc. (NORP), or Organization (ORG).

used for training neural ranking models. Importantly, it also has been shown to effectively transfer relevance signals to other collections (Nogueira and Lin, 2019), making it suitable for use with DTPs, which may include text from other domains.

We build MMPs and TMPs using the TREC Deep Learning 2019 passage dataset (Craswell et al., 2019) and the ANTIQUE passage ranking dataset (Hashemi et al., 2020). TREC DL uses the MS-MARCO passage collection and has 43 queries with deep relevance judgments (on average, 215 per query). The judgments are graded as highly relevant (7%), relevant (19%), topical (17%), and non-relevant (56%), allowing us to make more fine-grained comparisons. We use the test subset of ANTIQUE, which contains 200 queries with 33 judgments per query. These judgments are graded as convincing (20%), possibly correct (18%), on-topic (37%), and off-topic (25%). We opt to perform our analysis in a passage ranking setting to eliminate effects of long document aggregation—which is challenging for some neural models given a maximum sequence length in the underlying model—given that this is an area with many model varieties that is still under active investigation (Li et al., 2020).

3.2 Models

We compare a sample of several models covering a traditional lexical model (BM25), a conventional learning-to-rank approach (LightGBM), and neural models based on contextualized language models. We include two models that focus on query-time computational efficiency, and two representative models that use dense retrieval. The neural models represent a sample of the recent state-of-the-art ranking models. For each model, we provide the MRR (minimum relevance of 2) performance on the TREC DL 2019 passage benchmark when re-ranking the provided candidate passages.

BM25. We use the Terrier (Ounis et al., 2006) implementation of BM25 with default parameters. BM25 is an unsupervised model that incorporates the lexical features of term frequency (TF), inverse document frequency (IDF), and document length. (TREC DL 2019 MRR: 0.627.)

WMD. As a second unsupervised model, we use the Word Mover's Distance (Kusner et al., 2015) over (non-contextualized) GloVe (Pennington et al., 2014) embeddings (`glove-wiki-gigaword-100`). We use the implementation from the Gensim (Rehurek and Sojka, 2011) Python package. (TREC DL 2019 MRR: 0.364.)

SBERT. As an unsupervised model based on a contextualized language model, we use SBERT's (Reimers and Gurevych, 2019) pre-trained Bi-encoder model, trained on Semantic Textual Similarity, Natural Language Inference, and Quora Duplicate Question Detection data in multiple languages.[10] This approach has been shown by Litschko et al. (2021) to effectively perform cross-lingual retrieval. (TREC DL 2019 MRR: 0.465.)

LGBM (Ke et al., 2017). As a non-neural learning-to-rank baseline, we use the Light Gradient Boosting Machine model currently used by the Semantic Scholar search engine (Feldman, 2020).[11] This public model was trained on click-through data from this search engine, meaning that it services various information needs (e.g., navigational and topical queries). Not all of the model's features are available in our setting (recency, in-links, etc.), so we only supply the text-based features like lexical overlap and scores from a light-weight language model (Heafield et al., 2013). (TREC DL 2019 MRR: 0.580.)

VBERT (Devlin et al., 2019). We use a BERT model that uses a linear ranking layer atop a BERT pretrained transformer language model (Nogueira and Cho, 2019; MacAvaney et al., 2019a; Dai and Callan, 2019b). (This setup goes by several names in the literature, including Vanilla BERT (VBERT), monoBERT, BERT-CAT, etc.) We fine-tune the `bert-base-uncased` model for this task using the official training sequence of the MS-MARCO passage ranking dataset. (TREC DL 2019 MRR: 0.809.)

T5 (Raffel et al., 2020). The Text-To-Text Transformer ranking model (Nogueira and Lin, 2019) scores documents by predicting whether the concatenated query, document, and control tokens is likely to generate the term 'true' or 'false'. We use the models released by the authors, which were tuned on the MS-MARCO passage ranking dataset. We test both the `t5-base` (T5-B) and `t5-large` (T5-L) models to gain insights into the effect of model size. (TREC DL 2019 MRR: 0.868 (T5-B), 0.857 (T5-L).)

[10] `distilbert-multilingual-nli-stsb-quora-ranking`.
[11] `https://github.com/allenai/s2search`.

EPIC (MacAvaney et al., 2020). This is an efficiency-focused BERT-based model, which separately encodes query and document content into vectors that are the size of the source lexicon (where each element represents the importance of the corresponding term in the query/document). We use the `bert-base-uncased` model, and tune the model for ranking using the train split of the MS-MARCO passage ranking dataset with the code released by the EPIC authors with default settings. (TREC DL 2019 MRR: 0.809.)

DT5Q (Nogueira and Lin, 2019). The T5 variant of the Doc2Query model (DT5Q) generates additional terms to add to a document using a T5 model. The expanded document can be efficiently indexed, boosting the weight of terms likely to match queries. We use the model released by the authors, which was trained using the MS-MARCO passage training dataset. For our probes, we generate four queries to add to each document. As was done in the original paper, we use BM25 as a scoring function over the expanded documents. (TREC DL 2019 MRR: 0.692.)

ANCE (Xiong et al., 2021). This is a representation-based dense retrieval model that is trained using a contrastive learning technique. It is designed for single-stage dense retrieval. We use the model weights released by the original authors, which is based on the RoBERTa (Liu et al., 2019b) base model. (TREC DL 2019 MRR: 0.852.)

ColBERT (Khattab and Zaharia, 2020). This is a two-stage dense retrieval approach that uses multiple representations for each document (one per WordPiece token). It makes use of both a first-stage approximate nearest neighbor search to find candidate documents and a re-ranking stage to calculate the precise ranking scores. It is based on the `bert-base-uncased` model. We use the model weights released by the original authors. (TREC DL 2019 MRR: 0.873.)

3.3 Choosing δ

Recall that δ indicates the minimum absolute difference of scores in a document pair probe to have a positive or negative effect. Because each model scores documents on a different scale, we empirically choose a δ per model. We do this by re-ranking the official set from TREC DL 2019. Among the top 10 results, we calculate the differences between each adjacent pair of scores (i.e., $\{R(q,d_1) - R(q,d_2), R(q,d_2) - R(q,d_3), \ldots, R(q,d_9) - R(q,d_{10})\}$, where d_i is the ith highest scored document for q). We set δ to the median difference. By setting the threshold this way, we can expect the differences captured by the probes to have an effect on the final ranking score *at least* half the time. We explore this further in Section 4.1. Note that choosing a constant δ over one that is assigned per-query allows for testing probes where a complete corpus is not available, as is the case for some DTPs.

3.4 Significance Testing

We use a two-sided paired *t*-test to determine the significance (pairs of $R(q,d_1)$ and $R(q,d_2)$). We use a Bonferroni correction over each table to correct for multiple tests, and test for $p < 0.01$.

3.5 Software and Libraries

We use the following software to conduct our experiments: PyTerrier (Macdonald et al., 2021), OpenNIR (MacAvaney, 2020), `ir_datasets` (MacAvaney et al., 2021), Transformers (Wolf et al., 2019), sentence-transformers (Reimers and Gurevych, 2019), Anserini (Yang et al., 2018), and Gensim (Rehurek and Sojka, 2011).

4 Results and Analysis

We present results for MMPs in Table 1, TMPs in Table 2, and DTPs in Table 3 and highlight our key findings in the order they appear in the tables.

Contextualized language models can distinguish relevance grades when TF is held constant. From Table 1, we see that SBERT, VBERT, EPIC, T5, ColBERT, and ANCE all are able to distinguish relevance when term frequency is constant with at least a score of +0.18 across both datasets. Perhaps surprisingly, this is even true for our transfer SBERT model, which is not trained on relevance ranking data. These results are in contrast with models that score lexically (BM25, LGBM, and DT5Q), which score at most +0.10. The contextualized language models also perform better at distinguishing relevance grades than the other models when length and overlap are held constant, though by a lesser margin. When controlling for model type, it appears that the model's size is related to its effectiveness in this setting: The large version of T5 (T5-L, +0.53) performs better the base model (T5-B, +0.43).

Variable	Control	BM25	WMD	SBERT	LGBM	DT5Q	VBERT	EPIC	T5-B	T5-L	ColBERT	ANCE	Samples
	TREC DL 2019												
Relevance	Length	+0.40	+0.27	+0.43	+0.40	+0.48	+0.58	+0.54	+0.61	+0.66	+0.61	+0.53	19676
	TF	−0.03	+0.11	+0.25	+0.04	+0.10	+0.34	+0.27	+0.43	+0.53	+0.47	+0.45	31619
	Overlap	+0.41	+0.15	+0.39	+0.34	+0.47	+0.55	+0.50	+0.61	+0.65	+0.60	+0.49	4762
Length	Relevance	−0.05	−0.10	−0.01	+0.04	−0.07	*−0.01	−0.08	+0.01	+0.00	*+0.00	+0.00	515401
	TF	−0.14	−0.08	*+0.02	+0.02	−0.09	−0.09	−0.15	+0.01	*−0.00	*+0.03	+0.06	88582
	Overlap	+0.51	*+0.02	+0.15	+0.26	+0.24	+0.20	+0.11	+0.19	+0.18	+0.18	+0.15	3963
TF	Relevance	+0.88	+0.49	+0.34	+0.50	+0.73	+0.41	+0.48	+0.38	+0.42	+0.39	+0.35	303058
	Length	+1.00	+0.65	+0.46	+0.59	+0.84	+0.54	+0.61	+0.51	+0.53	+0.53	+0.47	19770
	Overlap	+0.79	*+0.02	+0.18	+0.37	+0.36	+0.26	+0.17	+0.26	+0.24	+0.25	+0.19	2294
Overlap	Relevance	+0.70	+0.47	+0.22	+0.20	+0.52	+0.19	+0.25	+0.17	+0.18	+0.14	+0.18	357470
	Length	+0.75	+0.59	+0.32	+0.35	+0.59	+0.31	+0.35	+0.28	+0.29	+0.27	+0.30	20819
	TF	+0.88	+0.25	*−0.00	−0.03	+0.47	+0.11	+0.17	+0.04	+0.06	*+0.03	+0.04	13980
	ANTIQUE												
Relevance	Length	−0.17	*−0.09	+0.12	−0.15	−0.09	+0.23	*−0.01	+0.26	+0.35	+0.13	+0.24	2257
	TF	−0.07	*+0.01	+0.18	*+0.02	+0.04	+0.23	+0.23	+0.34	+0.46	+0.28	+0.33	5586
	Overlap	*−0.01	*+0.00	+0.26	*+0.03	+0.12	+0.39	+0.16	+0.42	+0.47	+0.31	+0.36	1211
Length	Relevance	+0.04	*−0.07	+0.13	+0.23	+0.02	−0.07	+0.22	+0.12	+0.17	+0.17	+0.23	36164
	TF	−0.47	*−0.09	+0.12	+0.04	−0.23	−0.13	+0.25	+0.03	+0.19	+0.15	+0.24	8296
	Overlap	+0.67	*+0.07	+0.17	+0.33	+0.34	*+0.04	+0.35	+0.12	+0.17	+0.21	+0.28	902
TF	Relevance	+0.69	*+0.23	+0.37	+0.57	+0.56	+0.24	+0.53	+0.38	+0.42	+0.46	+0.45	19900
	Length	+1.00	*+0.48	+0.50	+0.68	+0.84	+0.39	+0.59	+0.36	+0.31	+0.55	+0.40	1397
	Overlap	+0.92	*+0.06	+0.14	+0.36	+0.45	*+0.08	+0.35	+0.15	+0.22	+0.25	+0.28	553
Overlap	Relevance	+0.42	*+0.29	+0.09	+0.01	+0.35	+0.21	*+0.01	+0.07	*+0.03	+0.04	*−0.01	27539
	Length	+0.67	*+0.33	+0.22	+0.35	+0.48	+0.10	+0.25	+0.13	*+0.08	+0.20	+0.18	1224
	TF	+0.87	*+0.21	+0.07	−0.05	+0.44	+0.14	−0.13	−0.01	−0.07	*−0.00	−0.08	4498

Table 1: Results of Measure and Match Probes (MMPs) on the TREC DL 2019 and ANTIQUE datasets. Positive scores indicate a preference towards a higher value of the variable. Scores marked with * are not statistically significant (see Section 3.4).

Probe	Dataset	BM25	WMD	SBERT	LGBM	DT5Q	VBERT	EPIC	T5-B	T5-L	ColBERT	ANCE	Samples
Rem. Stops/Punct	DL19	*+0.00	*−0.09	−0.23	−0.20	−0.04	+0.18	−0.78	−0.74	−0.80	−0.68	−0.59	9259
	ANT	*+0.04	*−0.19	−0.38	−0.24	−0.07	−0.25	−0.78	−0.64	−0.81	−0.74	−0.70	6540
Lemmatize	DL19	+0.00	−0.18	+0.05	−0.02	*+0.01	−0.04	−0.25	−0.42	−0.44	−0.38	−0.31	9259
	ANT	+0.04	*−0.01	−0.04	−0.09	+0.00	−0.22	−0.25	−0.30	−0.47	−0.25	−0.31	6392
Shuf. Words	DL19	*+0.00	−0.21	−0.06	−0.25	−0.11	−0.38	−0.76	−0.65	−0.76	−0.76	−0.40	9260
	ANT	*+0.04	*−0.11	−0.10	−0.25	−0.13	−0.61	−0.67	−0.65	−0.75	−0.67	−0.58	6545
Shuf. Sents.	DL19	*−0.00	*−0.01	−0.06	*−0.00	*−0.02	−0.13	−0.19	−0.20	−0.14	−0.14	−0.10	7290
	ANT	*−0.00	*−0.02	−0.04	*−0.00	*−0.02	−0.17	−0.20	−0.22	−0.22	−0.13	−0.14	4211
Shuf. Prepositions	DL19	+0.01	−0.21	−0.02	−0.02	*+0.02	−0.01	−0.11	−0.28	−0.31	−0.18	−0.24	9239
	ANT	+0.05	*−0.11	−0.04	−0.03	+0.01	−0.12	−0.16	−0.30	−0.36	−0.18	−0.29	6186
Typos	DL19	−0.23	−0.17	*+0.07	−0.15	−0.18	−0.09	−0.50	−0.37	−0.27	−0.42	−0.20	8982
	ANT	−0.32	*−0.41	−0.09	−0.27	−0.27	−0.40	−0.45	−0.38	−0.40	−0.56	−0.36	5551
+ DocT5Query	DL19	+0.34	+0.45	+0.33	+0.41	+0.15	−0.22	−0.63	−0.54	−0.60	−0.50	−0.47	9260
	ANT	+0.34	*+0.14	+0.17	+0.32	*+0.03	−0.42	−0.13	−0.67	−0.68	*−0.10	−0.37	6589
+ Non-Rel Sent.	DL19	−0.03	−0.10	+0.34	+0.20	+0.04	+0.26	+0.11	+0.33	+0.27	+0.33	+0.39	9260
	ANT	+0.25	*+0.04	+0.38	+0.31	+0.25	+0.08	+0.47	+0.28	+0.30	+0.34	+0.41	6346

Table 2: Results of Text Manipulation Probes (TMPs) on the TREC DL 2019 and ANTIQUE datasets. Positive scores indicate a preference for the manipulated document text; negative scores prefer the original text. Scores marked with * are not statistically significant (see Section 3.4).

Models generally have similar sensitivity to document length, TF, and overlap on TREC DL 2019. With the exception of models that use BM25 for scoring (BM25 and DT5Q), all the models we explore have similar behaviors when varying length, TF, and overlap. This suggests that although signals like TF are not *required* for EPIC, BERT, and T5 to rank effectively, they still remain an important signal when available. There are bigger differences between models when exploring the ANTIQUE dataset, suggesting differences in the models' capacity to generalize. We note that some of the largest differences relate to the relevance measurement, highlighting the differences in label definitions between the two datasets.

Probe	Dataset	BM25	WMD	SBERT	LGBM	DT5Q	VBERT	EPIC	T5-B	T5-L	ColBERT	ANCE	Samples
Paraphrase	MSPC	0.60	*0.89	0.94	*0.00	0.76	0.82	0.65	0.91	0.88	0.85	0.90	3421
Fluency	JFLEG	+0.03	*−0.07	*+0.00	*−0.00	*+0.02	+0.10	+0.22	+0.14	+0.07	+0.24	+0.17	5073
	(spellchecked)	*+0.01	*+0.05	*−0.03	*−0.00	*−0.01	+0.07	+0.20	+0.14	+0.13	+0.18	+0.09	5187
Formality	GYAFC	−0.03	*−0.03	−0.15	−0.09	−0.07	−0.05	+0.16	*+0.01	+0.15	−0.07	*+0.05	6721
	- entertain.	+0.04	*+0.01	*−0.11	−0.04	−0.01	*+0.04	+0.19	*+0.11	+0.23	*+0.01	+0.08	2960
	- family	−0.08	*−0.05	−0.18	−0.13	−0.11	−0.12	*+0.13	−0.08	*+0.08	−0.14	*+0.03	3761
Simplicity	WikiTurk	+0.13	*+0.21	+0.07	*−0.00	+0.05	*−0.03	*−0.01	−0.08	−0.13	+0.01	*−0.03	17849
Neutrality	WNC	+0.31	*+0.34	+0.11	*+0.00	+0.13	+0.11	+0.07	−0.00	+0.03	+0.13	−0.00	178252
Succinctness	XSum	+0.66	*+0.91	+0.58	+0.18	+0.66	+0.49	+0.18	−0.09	+0.07	+0.33	+0.47	17938
	CNN	+0.37	*+0.74	*+0.02	−0.43	+0.41	+0.16	−0.72	−0.58	−0.54	−0.33	−0.28	7154
	Daily Mail	*−0.01	+0.54	−0.37	−0.80	+0.06	−0.26	−0.93	−0.63	−0.58	−0.71	−0.56	18930
Factuality	NQ: PER	*−0.00	+0.16	−0.02	−0.00	−0.02	*−0.02	−0.07	+0.10	+0.14	+0.04	+0.04	72983
	NQ: GPE	*−0.00	+0.22	+0.02	+0.00	*+0.00	+0.09	+0.00	+0.27	+0.30	+0.22	+0.12	33528
	NQ: LOC	*−0.03	+0.21	*−0.12	*−0.02	*−0.02	*+0.01	*+0.02	+0.28	+0.29	+0.14	*+0.11	962
	NQ: NORP	+0.02	+0.30	+0.01	+0.01	+0.03	+0.07	+0.07	+0.25	+0.33	+0.26	+0.10	4250
	NQ: ORG	+0.01	+0.34	+0.01	*+0.00	*+0.01	+0.07	−0.01	+0.33	+0.38	+0.19	+0.13	13831

Table 3: Results of Dataset Transfer Probes (DTPs). The paraphrase probe is unsigned, as it is symmetric. Positive scores indicate a preference for fluent, formal, simplified, neutral (non-biased), succinct, and factual text. Scores marked with * are not statistically significant (see Section 3.4).

Trained contextualized language models are adversely affected by heavily destructive pre-processing steps. From Table 2, we find that removing stopwords and punctuation, performing lemmatization, and shuffling words negatively impacts most models across both datasets. Perhaps this is expected, given that this text is dissimilar to the text the models were pre-trained on. However, we note that the transfer SBERT model is far less affected by these operations, suggesting that these characteristics are not intrinsic to the contextualized language models, but rather a consequence of training them for relevance ranking. To gain further insights into the importance of word order, we control for local word order by only shuffling sentence order. We see that an effect remains for the contextualized models, though it is substantially reduced. This suggests that discourse-level signals (e.g., what topics are discussed earlier in a document) have some effect on the models, or the models encode some positional bias (e.g., preferring answers at the start of documents). To understand if the word usage of particular terms is important (rather than overall coherence), we also try shuffling only the prepositions in the sentence. We find that this has an effect on some models (most notably, both T5 models and ANCE), but not other models, suggesting that some end up learning that although these terms have meaning in the text, they are often unimportant when it comes to ranking.

Lexical models handle typographical errors better than trained contextualized language models. In all but one case (ANCE DL19), BM25, LGBM, and DT5Q are negatively affected by typographical errors less than the trained contextualized language models. This is a surprising result, given that contextualized language models should be able to learn common misspellings and treat them similarly to the original words (the transfer SBERT model largely ignores typos). This problem is particularly apparent for EPIC and ColBERT, which perform matching on the WordPiece level.

Trained contextualized models behave unexpectedly when additional content is introduced in documents. We find that models that rely heavily on unigram matching (e.g., BM25) and the transfer SBERT model respond positively to the addition of DocT5Query terms. Even the DocT5Query model itself sees an additional boost, suggesting that weighting the expansion terms higher in the document may further improve the effectiveness of this model. However, the contextualized models often respond *negatively* to these additions. We also find that adding non-relevant sentences to the end of relevant documents often increases the ranking score of contextualized models. This is in contrast with models like BM25, in which the scores of relevant documents decrease with the addition of non-relevant information. From the variable length MMPs, we know that this increase in score is likely not due to increasing the length alone. Such characteristics may pose a risk to ranking systems based on contextualized models, in which content sources could aim to increase their ranking simply by adding non-relevant content to their documents.

Paraphrasing text can drastically change ranking scores. In Table 3, we observe high scores across most models for the paraphrase probe. For BM25, this is because the document lengths differ to a substantial degree. Contextualized models—which one may expect to handle semantic equivalences like these well—assign substantially different scores for paraphrases up to 94% of the time. To dig into specific stylistic differences that could explain the paraphrase discrepancies, we explore fluency, formality, simplicity, and neutrality. We find that fluency and formality have a greater effect than simplicity and neutrality. Most notably, EPIC and ColBERT prefer fluent text with scores of $+0.18$ to $+0.24$, while lexical models have low or insignificant differences. Meanwhile, EPIC and T5-L prefer formal text, while ColBERT and T5-B either prefer informal text or have no significant differences. Finally, the largest preferences observed for simple and neutral text are from BM25—which are likely a consequence of reduced document lengths.

Model behaviors vary considerably with succinctness. First, BM25 has a strong (+0.66) preference for the summaries in XSum, a moderate preference for summaries in CNN (+0.37), and no significant preference for Daily Mail. This suggests different standards among the various datasets—for example, XSum (BBC) must use many of the same terms from the titles in the summaries, and provide long documents (reducing the score) that may not repeat terms from the title much. WMD also appears to be heavily affected by summaries, though in two of the three probes, there is insufficient evidence to claim significance. The preference for summaries in XSum can be seen across all models except T5-B, which very slightly favors the full text. Although most contextualized models prefer the full text for CNN and Daily Mail, VBERT prefers summaries for CNN ($+0.16$) while it prefers full text for Daily Mail (-0.26). Such discrepancies warrant exploration in future work.

WMD, T5, and ColBERT are biased towards factual answers. From our factuality probes, we see that most models have little preference for factual passages. However, WMD, both T5 variants, and ColBERT are biased towards answers that contain factually correct information. For T5 and ColBERT, this suggests that these models both learn some real-world information (likely in pre-training), and use this information as a signal when ranking. The larger size of T5-L appears to equip it with more knowledge, particularly about people, nationalities, and organizations. Curiously, although ColBERT exploits this information, the VBERT model (which uses the same base language model) does not appear to learn to use this information. For WMD, which doesn't have nearly the modeling capacity of T5 and ColBERT, the preference for factual information must be due to the fact that the word embeddings of the question are more similar to the word embeddings from the factual phrase than to those of the non-factual phrase. Although the contextualized language models should have the capacity to learn these trends and make similar decisions, this would be subject to such trends being present and distinguishable during fine-tuning. This suggests that using WMD over contextualized word embeddings may also improve the capacity of models to select factual answers.

4.1 Effect of δ

Recall that δ defines the model-specific threshold at which a difference in ranking score is considered important. To test the importance of the selection of δ, we test all probes while varying this parameter. Because the suitable values depend upon the range of scores for the particular ranker, we select δ by percentile among differences in the top 10 scoring passages of TREC DL 2019. Figure 2 provides a representative sample of these plots. We find that for low percentiles (corresponding to settings where minute changes in score are considered important), the scores and rankings of systems can sometimes be unstable (e.g., see BM25 and DT5Q in (c)). This suggests that there are variations of the score distributions close to 0. However, we remind the reader that such differences are unlikely to have impactful changes in a real ranked list. We find that by the 50th percentile of δ (i.e., the value we use for our experiments), the rankings of the systems produced by ABNIRML are generally stable. In most cases, the scores are stable as well, though in some cases drifting occurs (e.g., (c)). With a large δ, nearly no differences are considered important. In (c), we observe that L-GBM has no sensitivity to the paraphrases present in the probe, regardless of δ. These observations validate our technique for choosing δ.

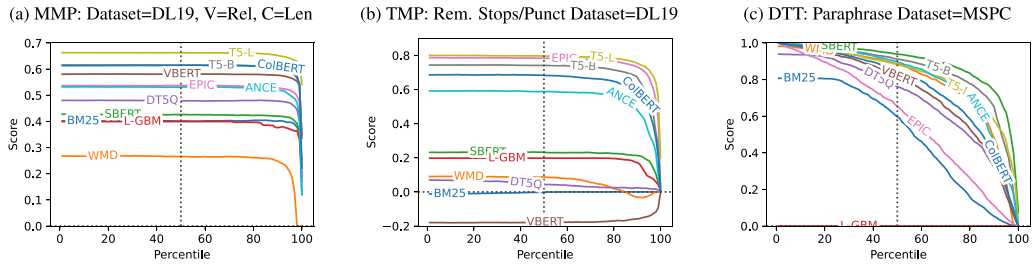

Figure 2: Plots of scores for three representative probes when varying δ to the specified percentile in TREC DL 2019. The vertical dashed line indicates the operational point of our experiments (the median value).

Probes	VBERT Stdev.	EPIC Stdev.
MMP	3.5	3.6
TMP	11.2	17.1
DTP	9.5	8.9

Table 4: Average standard deviations (square root of average variance) of 5 VBERT and EPIC models, by probe type.

4.2 Effect of Model Training

We observed that identical and similar base language models can differ in the behaviors they exhibit. To gain a better understanding of the origin of these differences, we probe 5 versions of the VBERT and EPIC models, each trained with different random seeds. We calculate the standard deviations of the performance over all the probes and report the average standard deviation for each probe type in Table 4. We find that among all probe types, MMPs are the most stable across random initializations and TMPs are the least stable. Curiously, the Stopword / punctuation removal TMP is the least stable probe across both models, with a stdev of 0.24 for VBERT and 0.27 for EPIC. In the case of VBERT, the probe score ranged from -0.33 to $+0.31$, highlighting that unexpected qualities can appear in models simply due to random variations in the training process. This is despite the fact that this probe is highly robust to the cutoff threshold on individual models (as seen in Figure 2(b)). Another probe with particularly high variance are the succinctness probe for VBERT using the CNN dataset, with a stdev of 0.23, and can either learn to prefer succinct ($+0.15$) or elaborative (-0.42) text, again due to the random initialization. These findings highlight that some biases can be introduced in the model training process randomly, rather than as a result of the pre-training process or model architecture.

5 Related Work

Pretrained contextualized language models are neural networks that are initially trained on language modeling objectives and are later fine-tuned on task-specific objectives (Peters et al., 2018). Well-known models include ELMo (Peters et al., 2018), BERT (Devlin et al., 2019), and T5 (Raffel et al., 2020). These models can effectively transfer signals to the task of ad-hoc retrieval, either by using the model directly (i.e., *vanilla* or *mono* models) (Nogueira and Cho, 2019) or by using the outputs as features into a larger model (MacAvaney et al., 2019a). There has been considerable work in this area; we refer readers to Lin et al. (2020) for a comprehensive survey on these techniques. We shed light on the mechanisms, strengths, and weaknesses of this burgeoning body of work.

Diagnostic datasets, proposed by Rennings et al. (2019), reformulate traditional ranking axioms—for example, that documents with a higher term frequency should receive a higher ranking score (Fang et al., 2004)—as empirical tests for analyzing ranking models. Rennings et al. studied neural ranking architectures that predate the rise of contextualized language models for ranking, and focused on just four axioms. Câmara and Hauff (2020) extended this work by adding five more previously proposed ranking axioms (e.g., term proximity [Tao and Zhai, 2007], and word semantics [Fang and Zhai, 2006]) and evaluating on a distilled BERT model. They found that the axioms are inadequate to explain the ranking effectiveness of their model. Völske et al. (2021) examine the extent to which these axioms, when acting in concert, explain ranking model decisions. Unlike these prior lines of work, we propose new probes that shed light onto possible sources of effectiveness, and test against current leading neural ranking architectures.

Although some insights about the effectiveness of contextualized language models for ranking have been gained using existing datasets (Dai and Callan, 2019b) and indirectly through various model architectures (Nogueira et al., 2019; Dai and Callan, 2019a; MacAvaney et al., 2020, 2019a; Hofstätter et al., 2020; Khattab and Zaharia, 2020), they only provide circumstantial evidence. For instance, several works show how contextualized embedding similarity can be effective, but this does not imply that vanilla models utilize these signals for ranking. Rather than proposing new ranking models, in this work we analyze the effectiveness of existing models using controlled diagnostic probes, which allows us to gain insights into the particular behaviors and preferences of the ranking models.

Outside of the work in IR, others have developed techniques for investigating the behavior of contextualized language models in general. Although probing techniques (Tenney et al., 2019) and attention analysis (Serrano and Smith, 2019) can be beneficial for understanding model capabilities, these techniques cannot help us characterize and quantify the behaviors of neural ranking models. CheckList (Ribeiro et al., 2020) and other challenge set techniques (McCoy et al., 2019) differ conceptually from our goals; we aim to characterize the behaviors to understand the qualities of ranking models, rather than provide additional measures of model quality.

6 Conclusion

We presented a new framework (ABNIRML) for analyzing ranking models based on three probing strategies. By using probes from each strategy, we demonstrated that a variety of insights can be gained about the behaviors of recently-proposed ranking models, such as those based on BERT and T5. Our analysis is, to date, the most extensive analysis of the behaviors of neural ranking models, and sheds light on several unexpected model behaviors. For instance, adding non-relevant text can increase a document's ranking score, even though the models are largely not biased towards longer documents. We also see that the same base language model used with a different ranking architecture can yield different behaviors, such as higher sensitivity to shuffling a document's text. We also find that some models learn to utilize real-world knowledge in the ranking process. Finally, we observe that some strong biases can appear simply by chance during the training process. This motivates future investigations on approaches for stabilizing training processes and avoiding the introduction of unwanted biases.

References

Matteo Alleman, J. Mamou, M. D. Rio, Hanlin Tang, Yoon Kim, and SueYeon Chung. 2021. Syntactic perturbations reveal representational correlates of hierarchical phrase structure in pretrained language models. arXiv, abs/2104.07578. https://doi.org/10.18653/v1/2021.repl4nlp-1.27

Arthur Câmara and Claudia Hauff. 2020. Diagnosing BERT with retrieval heuristics. In *ECIR*. https://doi.org/10.1007/978-3-030-45439-5_40

Daniel Fernando Campos, T. Nguyen, M. Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, L. Deng, and Bhaskar Mitra. 2016. MS MARCO: A human generated machine reading comprehension dataset. arXiv, abs/1611.09268.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2019. Overview of the TREC 2019 deep learning track. In *TREC*.

Zhuyun Dai and J. Callan. 2019a. Context-aware sentence/passage term importance estimation for first stage retrieval. arXiv, abs/1910.10687.

Zhuyun Dai and J. Callan. 2019b. Deeper text understanding for ir with contextual neural language modeling. *SIGIR*. https://doi.org/10.1145/3331184.3331303

Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2019. CAsT 2019: The conversational assistance track overview. In *TREC*.

J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Hui Fang, T. Tao, and ChengXiang Zhai. 2004. A formal study of information retrieval heuristics. In *SIGIR*. https://doi.org/10.1145/1008992.1009004

Hui Fang, T. Tao, and ChengXiang Zhai. 2011. Diagnostic evaluation of information retrieval models. *ACM Transactions on Management Information Systems*, 29:7:1–7:42. https://doi.org/10.1145/1961209.1961210

Hui Fang and ChengXiang Zhai. 2006. Semantic term matching in axiomatic approaches to information retrieval. In *SIGIR '06*. https://doi.org/10.1145/1148170.1148193

Sergey Feldman. 2020. Building a better search engine for semantic scholar. Blog post.

Helia Hashemi, Mohammad Aliannejadi, Hamed Zamani, and Bruce Croft. 2020. ANTIQUE: A non-factoid question answering benchmark. In *ECIR*. https://doi.org/10.1007/978-3-030-45442-5_21

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *ACL*.

Sebastian Hofstätter, Markus Zlabinger, and A. Hanbury. 2020. Interpretable and time-budget-constrained contextualization for re-ranking. In *ECAI*.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Guolin Ke, Q. Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and T. Liu. 2017. LightGBM: A highly efficient gradient boosting decision tree. In *NIPS*.

O. Khattab and M. Zaharia. 2020. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *SIGIR*. https://doi.org/10.1145/3397271.3401075

Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. 2015. From word embeddings to document distances. In *ICML*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: a benchmark for question answering research. In *TACL*. https://doi.org/10.1162/tacl_a_00276

Canjia Li, Andrew Yates, Sean MacAvaney, Ben He, and Yingfei Sun. 2020. Parade: Passage representation aggregation for document reranking. arXiv, abs/2008.09093.

Jimmy Lin, Rodrigo Nogueira, and A. Yates. 2020. Pretrained transformers for text ranking: BERT and beyond. arXiv, abs/2010.06467.

Robert Litschko, Ivan Vuli'c, Simone Paolo Ponzetto, and Goran Glavavs. 2021. Evaluating multilingual text encoders for unsupervised cross-lingual retrieval. In *ECIR*. https://doi.org/10.1007/978-3-030-72113-8_23

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. Linguistic knowledge and transferability of contextual representations. In *NAACL-HLT*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. RoBERTa: A robustly optimized BERT pretraining approach. arXiv, abs/1907.11692.

D. Loureiro, Kiamehr Rezaee, Mohammad Taher Pilehvar, and José Camacho-Collados. 2020. Language models and word sense disambiguation: An overview and analysis. arXiv, abs/2008.11608.

Sean MacAvaney. 2020. OpenNIR: A complete neural ad-hoc ranking pipeline. In *WSDM*.

Sean MacAvaney, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, Nazli Goharian, and Ophir Frieder. 2020. Expansion via prediction of importance with contextualization. In *SIGIR*. https://doi.org/10.1145/3397271.3401262

Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. 2019a. CEDR: Contextualized embeddings for document ranking. In *SIGIR*.

Sean MacAvaney, Andrew Yates, Sergey Feldman, Doug Downey, Arman Cohan, and Nazli Goharian. 2021. Simplified data wrangling with ir_datasets. In *SIGIR*. https://doi.org/10.1145/3404835.3463254

Sean MacAvaney, Andrew Yates, Kai Hui, and Ophir Frieder. 2019b. Content-based weak supervision for ad-hoc re-ranking. In *SIGIR*. https://doi.org/10.1145/3331184.3331316

Craig Macdonald, Nicola Tonellotto, Sean MacAvaney, and Iadh Ounis. 2021. PyTerrier: Declarative experimentation in python from BM25 to dense retrieval. In *CIKM*.

R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *ACL*. https://doi.org/10.1145/3459637.3482013

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. JFLEG: A fluency corpus and benchmark for grammatical error correction. In *EACL*. https://doi.org/10.18653/v1/E17-2037

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization. In *EMNLP*. https://doi.org/10.18653/v1/D18-1206

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with BERT. arXiv, abs/1901.04085.

Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. 2020. Document ranking with a pretrained sequence-to-sequence model. arXiv, abs/2003.06713. https://doi.org/10.18653/v1/2020.findings-emnlp.63

Rodrigo Nogueira and Jimmy Lin. 2019. From doc2query to docTTTTTquery. Self-published.

Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction. arXiv, abs/1904.08375.

I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and C. Lioma. 2006. Terrier: A high performance and scalable information retrieval platform. In *Proceedings of ACM SIGIR'06 Workshop on Open Source Information Retrieval (OSIR 2006)*. https://doi.org/10.1007/978-3-540-31865-1_37

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *EMNLP*. https://doi.org/10.3115/v1/D14-1162

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL-HLT*. https://doi.org/10.18653/v1/N18-1202

Reid Pryzant, Richard Diehl Martinez, Nathan Dass, S. Kurohashi, Dan Jurafsky, and Diyi Yang. 2020. Automatically neutralizing subjective bias in text. In *AAAI*. https://doi.org/10.1609/aaai.v34i01.5385

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140).

Sudha Rao and J. Tetreault. 2018. Dear sir or madam, may I introduce the YAFC corpus: Corpus, benchmarks and metrics for formality style transfer. In *NAACL-HLT*. https://doi.org/10.18653/v1/N18-1012

Radim Rehurek and Petr Sojka. 2011. Gensim–Python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2).

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *EMNLP*. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1410

D. Rennings, Felipe Moraes, and C. Hauff. 2019. An axiomatic approach to diagnosing neural IR models. In *ECIR*. https://doi.org/10.1007/978-3-030-15712-8_32

Marco Túlio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with checklist. In *ACL*. https://doi.org/10.24963/ijcai.2021/659

Anna Rogers, O. Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *TACL*. https://doi.org/10.1162/tacl_a_00349

A. See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *ACL*. https://doi.org/10.18653/v1/P17-1099

Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable? In *ACL*. https://doi.org/10.18653/v1/P19-1282

Koustuv Sinha, Robin Jia, Dieuwke Hupkes, J. Pineau, Adina Williams, and Douwe Kiela. 2021. Masked language modeling and the distributional hypothesis: Order word matters pre-training for little. arXiv, abs/2104.06644. https://doi.org/10.18653/v1/2021.emnlp-main.230

T. Tao and ChengXiang Zhai. 2007. An exploration of proximity measures in information retrieval. In *SIGIR*. https://doi.org/10.1145/1277741.1277794

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *ACL*. https://doi.org/10.18653/v1/P19-1452

Michael Völske, A. Bondarenko, Maik Fröbe, Matthias Hagen, Benno Stein, Jaspreet Singh, and Avishek Anand. 2021. Towards axiomatic explanations for neural ranking models. arXiv, abs/2106.08019. https://doi.org/10.1145/3471158.3472256

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art natural language processing. arXiv, abs/1910.03771.

Patrick Xia, Shijie Wu, and B. Van Durme. 2020. Which *BERT? a survey organizing contextualized encoders. In *EMNLP*.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. arXiv, abs/2007.00808.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *TACL*, 4. https://doi.org/10.1162/tacl_a_00107

Peilin Yang, Hui Fang, and Jimmy Lin. 2018. Anserini: Reproducible ranking baselines using Lucene. *J. Data and Information Quality*, 10. https://doi.org/10.1145/3239571