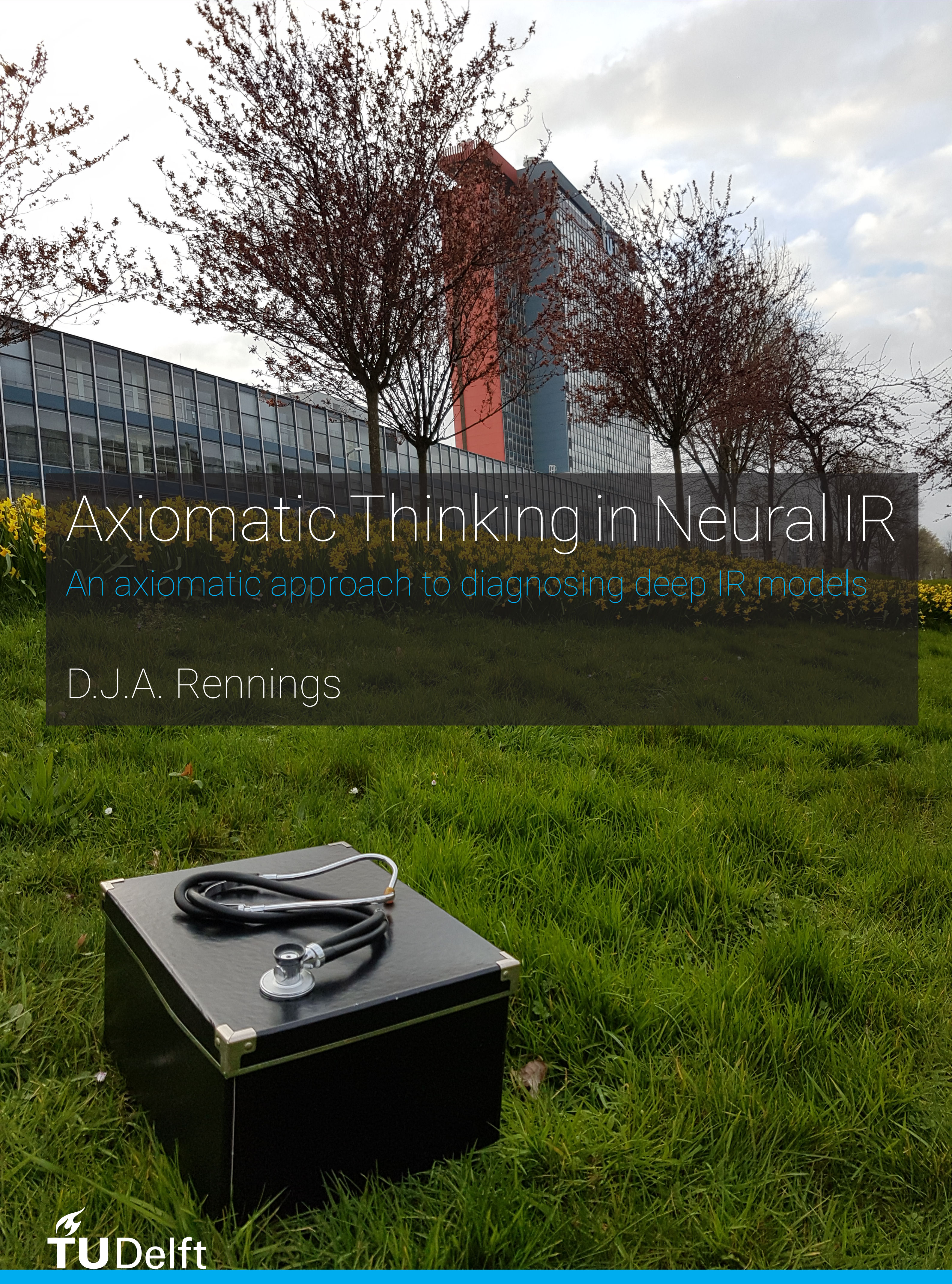


Axiomatic Thinking in Neural IR

An axiomatic approach to diagnosing deep IR models

D.J.A. Rennings

Axiomatic Thinking in Neural IR

An axiomatic approach to diagnosing deep IR models

by

D.J.A. Rennings

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday April 25, 2019 at 1:30 PM.

Thesis committee:	Dr. C. Hauff,	TU Delft, supervisor
	Prof. dr. ir. G. J. P. M. Houben,	TU Delft
	Dr. J. Urbano Merino,	TU Delft
	Dr. ir. C. C. S. Liem,	TU Delft
	F. Moraes MSc,	TU Delft, supervisor

An electronic version of this thesis is available at http://repository.tudelft.nl/.

Preface

Formally, a preface provides space for "... *acknowledgments to people who were helpful to the author during the time of writing*"[1], so let me start with that.

First and foremost, I would like to express my gratitude to my primary supervisor Claudia Hauff for her efforts to push our research to the max. I don't think the average student gets the opportunity to make a scientific contribution to the field of his or her thesis like I have been able to do thanks to your guidance. Thank you for supporting me throughout the entire process with response times that are not far behind modern search engines.

Next up is a word of thanks to Felipe Moraes, who has been acting as a second supervisor and has been a great technical go-to person as well as a wonderful sparring partner throughout my thesis work ("*You can put all formulas on this slide if you want, depending on how bored you want your audience to be...*").

I would also like to thank Geert-Jan Houben, Julián Urbano and Cynthia Liem for being part of my thesis committee. I would like to thank Naser Bakhshi for his guidance, open mind and the opportunity to present my work in front of business experts and Jerke Eisma and Edwin Wanner for their guidance and coordination.

Next to these persons, I would also like to thank a group of people that have been more than helpful before, during and (hopefully will be after) the time of writing. Steffie, graduating has been a long ride for the both of us and I want to thank you for your support, especially on the final mile. Mom and dad, I recall Dad saying: "*Het is hier geen hotel!*", well, it surely has been much more than that, thank you for your endless support.

As this work also concludes my stay at Delft University of Technology, I want to say a final thanks to Alexandru Iosup for his inspirational mentorship during my freshman year ("*Why go for a 9 if you can go for a 10?*") and Geert-Jan Houben for opening a door to an extraordinary visit overseas.

Daan Rennings
Cologne, April 2019

[1] See https://en.wikipedia.org/wiki/Preface.

Abstract

After surpassing human performance in the fields of Computer Vision, Speech Recognition and NLP, deep learning has been gaining scientific ground in IR. In spite of the sheer amount of publications that have proposed so-called neural IR approaches over the past decade, the field has not achieved the kind of progress seen in related fields. Over the past year or so, works have begun to solve the issues that complicate the progress of neural applications in IR. Among those issues we can find the lack of approaches to interpret and analyze neural IR models, which is addressed in this thesis.

We propose a novel approach to diagnose retrieval models that is rooted in the axiomatic approach to IR. Axioms encapsulate search heuristics that are expressed as constraints on retrieval functions. Existing axiomatic approaches have provided fruitful analyses of traditional IR models but are no longer viable to study neural IR models. Building forth on these approaches, we propose a novel approach to empirically analyze retrieval functions, suitable for neural models. Based on inspirations from the NLP and Computer Vision communities, we use model-agnostic diagnostic datasets in order to determine what kind of search heuristics models are able to learn. Since the creation of diagnostic datasets does not require a labeled dataset, we can apply the proposed pipeline to almost any dataset containing queries and documents.

We have shown for four specific axioms how to extend and relax them, in order to make them fit for obtaining diagnostic datasets. We have applied our diagnostic dataset creation pipeline to the `WikiPassageQA` and `MSMarco` corpora and evaluated three traditional baselines and six neural models. Our experiments on the `WikiPassageQA` dataset show that the proposed approach can indeed diagnose strengths and weaknesses of neural models. However, our experiments on the `MSMarco` dataset show that an axiomatic analysis based on the four axioms does not always diagnose factors that incur retrieval effectiveness. An interesting direction for future work is therefore to include more axioms in the diagnostic approach.

As possible extensions of the work carried out in this thesis, several roads of future work have been proposed. Among them, we can find reproducing experiments on other neural toolkits and employing the methodology on different IR tasks, but also researching the validity of axioms and adopting a specialized metric for axiomatic performance. We furthermore identified various opportunities to use diagnostic datasets beyond diagnosing neural models.

Concluding, we believe that the axiomatic approach to diagnosing neural IR models presented in this work is a step forward to gaining valuable insights into the black boxes that deep models are generally considered to be. We hope our work may prove a fruitful resource for analysis in the field of neural IR on the road towards achieving superior performance without losing sight of a better fundamental understanding of IR.

Contents

Symbols

$\texttt{Axiom}$	original axiom
$\overline{\texttt{Axiom}}$	extended, relaxed axiom
$\boldsymbol{C}$	collection language model
$\boldsymbol{d}$	document
$\boldsymbol{D}$	set of documents
$\boldsymbol{q}$	query
$\boldsymbol{Q}$	set of queries
w	word
y	relevance label
$\boldsymbol{Y}$	set of relevance labels
δ	parameter for document length difference (real value)
δ^*	relative parameter for document length difference (real value)
$\mathbb{N}$	number set of natural numbers, (0,1,2,...)

$abs(a)$	absolute value of a real number a
$avdl$	average amount of words per document
$c(w, \boldsymbol{q})$	count of word w in query $\boldsymbol{q}$
$df(w)$	number of documents containing word w
$idf(w)$	inverse document frequency of word w (here calculated as $\ln(1/(df(w)+1))$)
$g((\psi, \phi), \eta)$	evaluation function which computes the relevance score based on the feature representations
$L(S, \boldsymbol{d}, \boldsymbol{q}, y)$	loss for the score that scoring function S assigned to document $\boldsymbol{d}$ with relevance label y with respect to query $\boldsymbol{q}$
$p(c \mid \boldsymbol{C})$	probability of word w given by the collection language model $\boldsymbol{C}$
$S(\boldsymbol{d}, \boldsymbol{q})$	score assigned to document $\boldsymbol{d}$ with respect to query $\boldsymbol{q}$ as given by retrieval scoring function S
$S^{\star}$	optimal retrieval scoring function S
$\eta(\boldsymbol{d}, \boldsymbol{q})$	interaction function that extracts features from $\boldsymbol{d}$ and $\boldsymbol{q}$
$\phi(\boldsymbol{d})$	representation function that extracts features from document $\boldsymbol{d}$
$\psi(\boldsymbol{q})$	representation function that extracts features from query $\boldsymbol{q}$
$\lvert \boldsymbol{d} \rvert$	number of words in document $\boldsymbol{d}$

1 Introduction

Although the early works on deep learning date back to decades ago (e.g. [45, 46] and [61–64], as listed by Schmidhuber [113]), neither the term "deep learning" nor the approach was popular a decade ago [81]. Over the past decade, deep learning has acquired a lot of interest in both research and practice as a result of its excellent performance in fields such as Computer Vision, Speech Recognition and Natural Language Processing (NLP). Consequently, researchers have applied the technique to other fields with other types of data in the hope of achieving comparable superior performance over existing methods [34]. Among them we can also find the Information Retrieval (IR) community [85], whose work in this field has been termed as "neural IR" or "Neu-IR".

Long before the advent of neural IR, various approaches to retrieve information have been proposed. We will briefly introduce such approaches before we introduce neural IR, which allows us to look at the novel wave of deep approaches to IR from an abstract point of view.

Dating back to a century ago, researchers studied the *physical* process of retrieving books or papers, and patented mechanical and electro-mechanical devices as solutions over searching documents by hand [111]. However, such devices became obsolete with the advent of computers that could digitize this process. These early computer-based IR systems used so-called **boolean retrieval** in which a query (composed of search terms) was seen as a logical combination of words. In this approach, retrieval systems returned a set of documents that *exactly* matched the query, providing a basic means to search a document collection. In turn, such boolean models were surpassed by models that assigned a score to *each* document resembling its relevance to a query, returning a ranking of documents in a collection, known as **ranked retrieval** [50].

Following these developments, various works have introduced notions such as relevance feedback, query expansion, semantic matching, inverse document frequency and various vector space, probabilistic and language models. Among the introduced models, we can find Okapi BM25: a probabilistic model that is still a widely adopted baseline in today's neural IR works. Okapi BM25[1] was developed by Robertson et al. [107], who experimented with variations of the BIM (Binary Independence Model)[2] on various test collections. Their experiments first led to the BM11 and BM15 models that were ultimately combined in BM25.

As illustrated by the development of BM25, ranked retrieval functions were so far manually devised and tuned by hand through experimentation [111]. The seminal work by Fuhr [44] described the idea to automatically tune the ranking function for all queries for a particular document collection, the idea would become known as **learning to rank** (LTR). This approach however only became effective with the availability of more training data (e.g. web query logs) and methods that were able to handle a larger number of features [111]. However, these typically hand-crafted features (e.g. the query term frequency or the inverse document frequency of components of a document such as the title or body) were time-consuming to design and over-specific in definition [98].

Finally, subsequent to the learning to rank paradigm, the IR community has seen the emergence of **neural IR**. Whereas learning to rank still required the manual creation of features, neural IR approaches auto-

[1] "Okapi" is the name of the experimental text retrieval system at Robertson et al. [107] used at City University London and "BM" stands for Best Match, Okapi BM25 is often abbreviated as BM25.

[2] BIM, introduced in [106], was one of the first probabilistic models, to make estimations of probability feasible, it assumed that queries and documents can be represented as binary term vectors and that terms are independent [80].

matically learn which features to use from raw input through employing neural networks of multiple layers. Sometimes these approaches have achieved large improvements over previous state of the art, for example, [94] outperformed previous best results on two corpora with a large margin (>7.5 in MRR@10 on `MSMarco` and >18.5 in MAP on `TREC-CAR`). However, the neural approaches proposed so far suffer from a lack of model robustness to the corpus (i.e. performances are highly dependent upon the employed dataset). In addition, the neural approaches—different from traditional models (for which we know the retrieval formula it employs) and to a smaller extend learning to rank approaches (for which we know the features it employs and can obtain how important each feature is)—have so far contributed little to a better understanding of IR concepts: they have largely remained "black boxes" and have therefore received considerable criticism [49, 85]. Similar to advancements made in previous retrieval paradigms, work has now begun in the IR community to further research neural IR approaches, to possibly reap the benefits deep learning has offered in related fields.

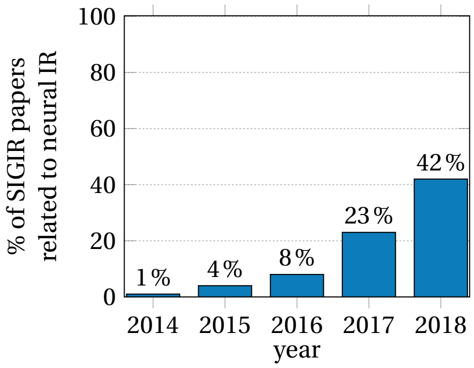


Figure 1.1: The percentage of papers related to neural IR at the ACM SIGIR[3] conference as determined by a manual inspection by Mitra et al.: a clear trend of growing popularity of the field. Figure adapted from [86].

1.1. Research Objective

Despite academic excitement surrounding the field [29] and consistent yearly increases in publications related to neural IR (see Fig. 1.1), the IR community has not enjoyed the kind of progress seen in other research areas such as NLP and Computer Vision. A number of issues that may have contributed to the slow progress of neural IR have been identified [28, 29, 85]. Overall, our community lacks:

1. Adequately large-scale publicly available datasets for training neural IR models;
2. Shared, centralized, code repositories of neural IR models;
3. Proper tools to interpret and analyze neural IR models.

We can find that some progress on these issues has been made over the past few years. For example, regarding the first issue, with the release of the large-scale datasets encompassing not the traditional several hundreds or a few thousands of instances with relevance labels (e.g. the TREC[4] `Robust` [125] and `QA` [128] datasets), but several thousands up to several hundreds of thousands (e.g. `WikiPassageQA` [22], `MSMarco` [92], `TREC-CAR` [37]). Regarding the second issue, a few neural IR toolkits that bundle multiple neural IR models have been established, such as (CO-)PACCR [60] and SERT [123] and the widely-used MatchZoo [38]. With regard to the third issue, some progress has been made in works aimed at interpreting neural IR models through studying components of trained deep neural networks [21, 98].

Building forth on multiple of these recent developments (i.e. by using large-scale public datasets and a shared neural IR repository), this thesis progresses on the third issue by providing a novel approach to analyze neural IR models. This approach aims to diagnose the strengths and weaknesses of neural IR models and is based upon so-called axiomatic thinking. In short, axiomatic thinking strategies test to what extent retrieval models adhere to retrieval heuristics. When applied to neural IR models, this may allow us to analyze their strengths and weaknesses. Hence, we adopt the following research question:

Research Question: How can we diagnose the strengths and weaknesses of neural IR approaches using axiomatic thinking?

[3] The Association for Computing Machinery (ACM) Special Interest Group for Information Retrieval (SIGIR), a leading conference in the field of IR.
[4] The Text REtrieval Conference, an ongoing series of workshops focusing on a list of different IR research areas, or tracks.

1.2. Approach

Our approach to address the research question (posed in Section 1.1), is introduced in this section. Since this approach is rooted in axiomatic thinking, we will first introduce this paradigm in Section 1.2.1. Subsequently, in Section 1.2.2, we detail how the research question will be addressed in this thesis.

1.2.1. The axiomatic thinking paradigm

Before the prevalence of deep nets in IR, Fang et al. [43] have studied a research question close to our first research question*"how to design a new evaluation methodology to help identify the strengths and weaknesses of retrieval functions"*. In [42], they first pointed out that previous works had attempted to identify an effective retrieval formula through extensive empirical experiments, which achieved abstract results with some retrieval formulas performing better under "some conditions". Hence, they tried to shed light on what these conditions might be. Motivated by the empirical observation that retrieval effectiveness is closely related to the use of various retrieval heuristics [42], they first defined several retrieval heuristics in a formal way, resulting in constraints expressed in a language similar to retrieval formulas. They then analyzed whether retrieval formulas fulfilled these constraints, which were later coined *axioms* [40]. Fang et al. [42] found that models' retrieval effectiveness was closely related to their fulfillment of the axioms and hence, the axioms provided a means to identify strengths and weaknesses of retrieval functions.

For example, Term Frequency Constraint 1 (known as `TFC1`) encapsulates the heuristic to favor a document with more occurrences of a query term [42]. Formally, it can be expressed as: let $\boldsymbol{q} = \{w\}$ be a single-term query and $\boldsymbol{d_1}$ and $\boldsymbol{d_2}$ be two documents of equal length, i.e. $|\boldsymbol{d_1}| = |\boldsymbol{d_2}|$. Further, let $c(w, \boldsymbol{d})$ be the count of word w in document $\boldsymbol{d}$ and $S(\boldsymbol{d}, \boldsymbol{q})$ be the retrieval status value (score) a retrieval function S assigns to $\boldsymbol{d}$ with respect to $\boldsymbol{q}$. `TFC1` then states that if $c(w, \boldsymbol{d_1}) > c(w, \boldsymbol{d_2})$ holds, $S(\boldsymbol{d_1}, \boldsymbol{q}) > S(\boldsymbol{d_2}, \boldsymbol{q})$ should also hold. The latter formulation can be used to analyze under what conditions a retrieval formula (expressed in the same language) adheres to the heuristic encapsulated in the axiom (as will be detailed in Section 2.5.1).

Next to the methodology for diagnosing IR models, Fang et al. [42] also proposed how, given the diagnosis, retrieval models could be improved in terms of retrieval effectiveness. For example, a manual analysis showed that the Okapi BM25 retrieval model fulfills `TFC1` under a specific condition (the query term w should not be present in more than half of the documents in the corpus), but can be modified to fulfill `TFC1` under any condition. As a result, the modified version of Okapi BM25 achieves higher retrieval effectiveness.

The adoption of such methods allowed the IR community to go from extensive empirical experiments that exhibit a trial-and-error methodology of testing the retrieval effectiveness of a range of models with various settings across different collections, to a search for strengths and weaknesses (and ways to fix weaknesses) of retrieval functions as guided by IR heuristics. The difference in obtained results between both approaches is illustrated in Table 1.1.

	Collection 1	*Collection* 2	⋯	*Collection j*
Model 1	0.36	0.38	…	0.56
Model 2	0.33	0.40	…	0.55
⋮	⋮	⋮	⋱	⋮
Model i	0.31	0.39	…	0.52

Conclusion: e.g., *Model* 1 performs best for *Collections* 1 and j, but *Model* 2 performs best for *Collection* 2.

	Axiom 1	*Axiom* 2	⋯	*Axiom n*
Model 1	*Yes*	*Yes*	…	C_3
Model 2	*Yes*	*Yes*	…	*Yes*
⋮	⋮	⋮	⋱	⋮
Model m	C_6	C_6	…	C_6

Conclusion: e.g., the retrieval effectiveness of *Model m* may be improved if it can be adapted to fulfill *Axiom* 1.

Table 1.1: Illustration of results obtained in extensive experiments adopted in works prior to [42] and results obtained with the axiomatic approach Fang et al. introduced. On the left, we display a subset of results presented in [110] and on the right, we display a subset of results obtained in [42] (for a different set of models). "Yes" means the model always fulfills the axiom, whereas "C_x" means the model fulfills the axiom under certain conditions and therefore does not always adhere to the heuristic.

The so-called "axiomatic thinking" approach in IR has since then been further extended with more axioms to further improve retrieval models and propose new models, as will be further discussed in Section 2.5.

1.2.2. Axiomatic thinking in the neural IR era

Nearly all works on axiomatic thinking have considered non-neural models, such as Okapi BM25. Nowadays, rather than being the model under study, traditional models have been adopted as baselines in papers that instead focus on introducing a novel neural IR model. Although rich of this new branch of *neural* models that sometimes achieve state of the art performance, the IR community is again facing difficulties in identifying a model's strengths and weaknesses (just like before the introduction of axiomatic thinking in IR [43]).

In this thesis, we propose to follow the axiomatic thinking methodology to establish a means to identify the strengths and weaknesses of neural IR models. Ideally, we would employ the traditional axiomatic approach encompassing analytical validation and direct retrieval formula adaptation. However, since deep learning models may contain millions (or billions) of parameters [85], analytical validation and manual retrieval formula adaptation are no longer feasible. As a solution, this work proposes the creation of so-called "diagnostic datasets" to mimic the axiomatic analysis process in an empirical setting. Each of the diagnostic datasets created in this work, can be used to diagnose the fulfillment of one axiom.

The use of diagnostic datasets was inspired by the fields of NLP and Computer Vision, where dataset creation for diagnostic purposes is an established approach [65, 66, 129], as will be further detailed in Section 2.4.2. In contrast to the de facto performance validation through standard test collections and evaluation metrics—which offers little insight into *why* one model achieves a better performance compared to another—the diagnostic methods allow one to obtain such knowledge at the granularity of the diagnostic dataset (individual axioms in our case).

Moving on from diagnosing to improving neural IR approaches, the diagnostic datasets may also be used to improve the retrieval effectiveness of neural models. In this work we have specifically proposed to utilize the fact that each diagnostic instance from a diagnostic dataset can directly be used as a training instance for neural IR models.

1.3. Scientific Contributions

With this work, we make several scientific contributions to the the field of IR by bringing the axiomatic thinking paradigm to the neural IR era. The main contribution of our work is **to showcase that a transformation from an analytical axiom to a diagnostic dataset is possible and offers us a new tool to diagnose retrieval models that are too complex to be analyzed theoretically**. On a more specific level, our contributions are as follows:

- We have presented a novel methodology for identifying strengths and weaknesses of IR models, specifically designed for neural IR models. As a basis for this diagnostic approach, we have proposed how one can convert established axioms to versions that can be used to obtain diagnostic datasets.
- Using the proposed diagnostic methodology, we have identified the strengths and weaknesses of several state-of-the-art neural IR models at the level of individual axioms.

In addition to the contributions encapsulated in this thesis, we have released all code that has been used in this work[5]. Finally, we note to the reader that part of this thesis has been featured in the 41st European Conference on Information Retrieval (ECIR 2019), under the title "An Axiomatic Approach to Diagnosing Neural IR Models" [104].

1.4. Thesis Outline

The remainder of this thesis is organized as follows. We start by providing readers with a background of IR and discuss related works in Chapter 2. Subsequently, in Chapter 3, we discuss our approach to obtaining diagnostic datasets. In Chapter 4, we then discuss experiments conducted to analyze neural IR models.Finally, in Chapter 5, we summarize and conclude our work and introduce a starting point for several roads of future work.

[5] See https://github.com/drennings/ADIR/.

2

Related Works

This chapter serves two purposes. First, it provides a *background* to make readers familiar with Information Retrieval (Section 2.1), learning to rank (Section 2.2) and neural IR (Section 2.3). Although we study neural IR models in this work, a brief introduction to learning to rank approaches allows us to nicely introduce the neural IR approaches and is therefore included in this chapter. Since good knowledge sources exist for both IR and neural IR[1], we here stay at a level of abstraction sufficient to comprehend the remainder of this thesis.

As a second purpose, this chapter provides as an overview of works that are *related* to this thesis, in the sense that they have researched topics close to ours. We elaborate upon two strands of such related works: the current adopted means for evaluating deep learning approaches (Section 2.4) and works on axiomatic thinking in IR (Section 2.5).Towards the end of each of these sections, we elaborate upon how our research relates to the discussed works.

2.1. Information Retrieval

IR typically involves a user that has a certain information need which (s)he translates into a query. Generally, this query serves as input for a retrieval engine that returns ranked documents to the user. Traditionally, ranked retrieval systems have employed an index of a document collection and assigned a relevance score (with regard to the query) to each of the documents in the collection. A graphical overview of this general process is depicted in Fig 2.1. A common example of this process is the classical use of a commercial web search engine (e.g. Google, Bing, Yahoo, Baidu) in which a user types in a query which is executed on a retrieval system that—from an index of webpages on the World Wide Web—returns ranked webpages (the documents in this example) to the user.

[1] We recommend the works of Manning et al. [80] and Baeza-Yates et al. [9] for an introduction to IR and the comprehensive materials of Mitra and Craswell [85, 86] and Guo et al. [49] on neural models for IR, as well as a full day tutorial at WSDM and a keynote at Microsoft Research on neural IR.

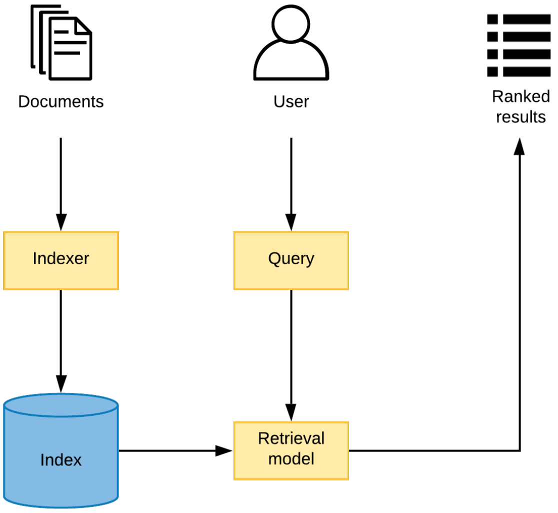


Figure 2.1: Graphical overview of the typical IR process, inspired by [50].

2.1.1. IR types and tasks

Over time, the retrieval of information has taken various forms. For example, in the past few decades, image and video retrieval - the retrieval of images or videos from a collection of images or videos - have come to life [32]. In more recent years, voice search - in which users speak their queries rather than type them - has gained attention [51]. In this thesis we however only consider **text retrieval**: a central form of retrieval in IR [49], in which both the input and output of information retrieval process consists of natural language.

Next to these types of retrieval, the field of information retrieval can be categorized into different *tasks* or subtopics, such as ad-hoc retrieval, text summarization, question answering and more novel tasks such as complex answer retrieval [91]. In this thesis, we focus on the ad-hoc retrieval and question answering tasks, which are further detailed next.

Ad-hoc retrieval typically considers the task of finding those documents from a large document collection that are relevant to a user's query [122]. Originally, ad-hoc retrieval considered searching news reports and government documents [86], but the most popular example nowadays is web search [9]. The term *ad-hoc* refers to the scenario where the documents in the collection remain relatively static [49]. In ad-hoc retrieval, a user's query can consist of a set of only a few keywords up to several, whereas search engine queries tend to be at the shorter end of the range [86]. These queries may specify an *ill-defined* information need. For instance, users that pose the query "BBC" are probably looking for the home page of the corporation, yet they expect the search engine to infer that specific information request from the three letters they entered [112]. Moreover, the documents that a user is looking for typically differ from the search terms in length while they also come from a different author. Such *heterogeneity* can lead to critical vocabulary mismatch problems (i.e. not all query terms can be found in a relevant document and vice-versa), that have been addressed with semantic matching (i.e. matching words and sentences with similar meanings), although exact matching is indispensable especially with rare query terms [49].

Question answering (QA) is the IR task of returning a piece of text as an answer to a natural language question based on some information resources [49]. In question answering, the user is interested in a concise, comprehensible and correct answer, which may refer to a word, sentence, paragraph, or an entire document [69]. In contrast to ad-hoc retrieval, *questions* typically specify a *well-defined* information need and can carry more information than a few search keywords, as they represent syntactic and semantic relationships between the search terms [69]. For example, consider how the question “Who is the architect of the Hancock building in Boston” [69] differs from the aforementioned query “BBC”. Moreover, compared to the query and documents considered in ad-hoc retrieval, questions and answers considered in QA show *reduced heterogeneity* in terms of e.g. length [49]. Hence, QA typically requires less semantic matching, although vocabulary mismatch remains a basic problem in QA [49].

2.1.2. Traditional IR models

A distinction can be made between IR models that rank documents based on their relevance to a query, known as **query-dependent** models, and models that rank documents based on their own importance, known as query-independent models [76]. As hinted by the previous paragraphs, we only study the first category of models in this thesis.

Moreover, at the beginning of Chapter 1, we already briefly introduced several categories of retrieval approaches, among which boolean retrieval. Models that fall under this category are however, among other early models such as vector space models, hardly found in today's IR papers. Hence, we will in this section focus on *traditional* models that are included in papers within the field of neural IR.We will thereby focus on presenting their intuition, rather than their mathematical deduction. We will later introduce the learning to rank and neural IR approaches respectively in Section 2.2 and Section 2.3.

Probabilistic models

Responding to a call to a firmer theoretical footing (rather than empirical evidence) of and more explicit assumptions in IR models [80], the probabilistic approach was proposed. Models under this approach specifically modeled that a retrieval system must necessarily be dealing with probabilities, as no retrieval system can be expected to predict with certainty which documents are relevant (to a user) [82, 105]. However, probabilistic approaches did not consistently outperform other approaches until the BM25 model was proposed [80], already several decades ago. BM25 is displayed in Eq. 2.1.

Although the model was the result of empirical experiments, it is grounded in probabilistic arguments. For example, it follows the statistical query term weighting scheme of Robertson and Jones [106], that would become known as inverse document frequency (IDF). This IDF component is the leftmost component of the product in BM25 (as displayed in Eq. 2.1). It valuates the informativeness of a term by weighting it on the amount of documents in a document corpus of size N that contain the term (known as the document frequency of a term, $df(w)$). Moreover, BM25 accounts for the count of each query term in a document ($c(w,\boldsymbol{d})$), known as term frequency or TF). This TF is normalized by the length of the document ($|\boldsymbol{d}|$) divided by the average document length ($avdl$) and regulated by parameters k_1 and b as displayed in the middle of the product. Finally, the rightmost component of the product in BM25 accounts for how often a query term is present in a query ($c(w,\boldsymbol{q})$), know as the query term frequency, which is regulated by the k_3 parameter. Note that the parameters k_1, b and k_3 need to be set manually (for which ranges in which the model typically works well are known), whereas the values for statistics such as the document frequency of a word are obtained from the index employed by the model.

$$BM25(\boldsymbol{d},\boldsymbol{q}) = \sum_{w\in\boldsymbol{q}\cap\boldsymbol{d}} \left(\ln \frac{N-df(w)+0.5}{df(w)+0.5} \times \frac{(k_1+1)c(w,\boldsymbol{d})}{k_1(1-b+b\frac{|\boldsymbol{d}|}{avdl})+c(w,\boldsymbol{d})} \times \frac{(k_3+1)\times c(w,\boldsymbol{q})}{k_3+c(w,\boldsymbol{q})} \right) \tag{2.1}$$

Equation 2.1: Retrieval formula of BM25 [43].

Language modelling

BM25 uses query terms as inputs to heuristic components to directly estimate the probability of relevance of a document with respect to the query. The intuition behind language modelling, on the other hand, is to first estimates the likelihood of generating each query term by randomly sampling terms from document [86], hence it is generally referred to as query likelihood (QL). Secondly, the model employs the product of these likelihood estimates to estimate the probability of relevance of a document with respect to the query (under Bayes theorem, and the assumption that all documents have equal prior probabilities [86]). Most formulations of language modelling also employ some form of *smoothing* by sampling not only from the considered document but also from the complete document collection. Hence, they also model the likelihood of sampling a query term from the document collection (to avoid the issue of assigning probability 0 to documents that do not contain all query terms). Omitting a mathematical deduction, we here simply show the eventual retrieval formula for the query likelihood model with Dirichlet smoothing [79], in which μ is the smoothing parameter and $p(w|\boldsymbol{D})$ is the probability of sampling w from the document collection $\boldsymbol{D}$.

$$QL(\boldsymbol{d},\boldsymbol{q}) = \sum_{w\in\boldsymbol{q}\cap\boldsymbol{d}} \left(c(w,\boldsymbol{q}) \cdot \ln(1+\frac{c(w,\boldsymbol{d})}{\mu\cdot p(w|\boldsymbol{D})}) + |\boldsymbol{q}|\cdot \ln\frac{\mu}{|\boldsymbol{d}|+\mu} \right) \tag{2.2}$$

Equation 2.2: Retrieval formula of QL with Dirichlet Prior Smoothing [43].

Pseudo-relevance feedback

Pseudo-relevance feedback (PRF) methods, such as the Relevance Models (RM) proposed by Lavrenko and Croft [70], typically outperform the aforementioned models at the cost of executing an additional round of retrieval [86]. The models are based on the language modelling approach, but conduct two rounds of retrieval instead of one. The set of ranked document from the first round of retrieval is used to select *query expansion* terms to augment the original query, which is subsequently used to retrieve the final set of documents that is presented to the user. In this work we employ the RM3 model, which, different from the initial Relevance Models (RM1 and RM2) combines both the original and the expanded query[2] [1]. RM3 is often referred to as one of the most effective methods for automatic query expansion [23]

The query expansion component in RM3 allows it to deal with a vocabulary mismatch between a query and a document, which is plaguing the BM25 and QL models [86]. However, the RM3 model contains six parameters (e.g. for the amount of documents and terms to use for the pseudo-relevance feedback), whereas BM25 and QL respectively only have 3 and 1 parameter(s). These parameters can be tuned (i.e. optimized within a specific range of values to test) to specific tasks by running a model and evaluating it (which will be detailed in Section 2.4.1), which is thus a more tedious task for RM3.

2.1.3. Evaluations in IR

In IR, evaluation considers the process of assessing how well a system meets the information needs of its users [124]. Two broad classes of such assessments consider user-based and system evaluation. Looking at the purpose of evaluation, user-based evaluations—that measure the user's satisfaction with the system—are preferred over system evaluations—that focus on how well a system can rank documents according to some relevance judgments [124]. However, the expense and difficulty (consider e.g. reproducibility and reusability) of obtaining user assessments of a *system* have led IR researchers to primarily rely on the less expensive system evaluation[3]. These system evaluations, also known as offline- or batch evaluations, simulate a user-based evaluation through employing a test collection [112]. A classical test collection consists of [112]:

- a collection of **documents** with unique identifiers (docids);
- a set of **topics** (to which we will refer as queries[4]), also with unique identifiers (qids);
- a set of **relevance judgments** (qrels - query relevance set) typically obtained from human assessors that judge which documents are relevant to a given query.

This test-collection based approach is also known as the **Cranfield paradigm**, named after a seminal work by Cleverdon [17] at the Cranfield Aeronautical College. In this work, *all documents* in a collection were labeled, meaning that for each document it was specified whether it was relevant or not to every query. Since such labeling is typically done by human assessors, the costs of labeling all documents in a large collection for every query have led IR researchers to primarily rely on a different approach, known as **pooling** [118], as is done in for example TREC and NTCIR[5] [115].

If we use a test collection in conjunction with an evaluation measure (metric), we can compare the effectiveness of different approaches for the retrieval task at hand, and subsequently, employ a statistical test to obtain whether an approach is significantly better than another approach.We will now briefly introduce the evaluation measures employed in the experiments in this thesis, an explanation of the employed statistical tests is beyond the scope of this background section.

[2] The eventual retrieval formula of RM3 is not easily represented without presenting a mathematical background of the QL and PRF approach, as can be seen in e.g. [55, 77]. As we merely employ RM3 as a (strong) baseline while our focus is on studying neural approaches, we do not present the retrieval formula of RM3.

[3] Approaches that go beyond offline evaluation have gained more attention recently (think of e.g. the *real-time* Live QA and Summarization Tracks of TREC 2017), but are beyond the scope of this thesis.

[4] For completeness we note that topics are converted into queries by a retrieval system [115], so for example a topic "BBC" could be typed in by a user so that the retrieval system sees the three typed letters from the user (instead of e.g. a voice query in which a user speaks these characters). We do not make a distinction between the two here since the topics considered in this work are always equal to the queries.

[5] The NII Testbeds and Community for Information access Research, an ongoing series of evaluation workshops, a Japanese counterpart of TREC.

Metrics

A metric should reflect the users' satisfaction with the system, which largely depends upon the IR task. Hence, tens of metrics[6] have been proposed of which 3 have been employed in the experiments in this thesis. We will first introduce metrics at the level of a single test collection instance and then introduce metrics at the level of a complete test collection, which have been employed in this work.

Precision is a set-based metric: it is computed without taking a ranking into account. It simply equates to the fraction of relevant documents (abbreviated as docs) out of all documents returned by a retrieval model (see Eq. 2.3). The *P@k* (precision at k) metric is similar to the precision metric but only considers the top k documents that are returned by a ranking model. Note that this metric still does not take the *order* of (relevant and non-relevant) documents into account. A metric that does take this order into account, is the *AveP* (average precision) metric. This metric also takes the number of relevant documents R into account and contains an indicator function $rel(k)$ which is set to 1 if the document at position k is relevant and set to 0 otherwise. It is the average of the precision values obtained after each relevant document is retrieved in the n retrieved documents - if a relevant document is not retrieved, its precision is valuated as 0 (see Eq. 2.4). Another metric, the reciprocal rank (RR) metric, only considers the rank of the first relevant document (see Eq. 2.6).

$$\boldsymbol{precision} = \frac{\text{num. of relevant docs retrieved}}{\text{num. docs retrieved}} \tag{2.3}$$

$$\boldsymbol{P@k} = \frac{\text{num. of relevant docs retrieved}}{\text{num. docs retrieved}} \tag{2.4}$$

$$\boldsymbol{AveP} = \frac{\sum_{k=1}^{n} P@k \cdot rel(k)}{R} \tag{2.5}$$

$$\boldsymbol{RR} = \frac{1}{\text{rank of first relevant doc}} \tag{2.6}$$

Now, the metrics at the level of a *whole* test collection are defined by simply averaging the measures obtained per query, as displayed in Eq. 2.7 and Eq. 2.8. Note that $P@K$, when used as an evaluation metric in experiments typically refers to a metric at the collection level that similarly averages the scores of the metric (by taking the sum of the result per query and dividing it by $|\boldsymbol{Q}|$).

$$\boldsymbol{MAP} = \frac{1}{|\boldsymbol{Q}|} \sum_{\boldsymbol{q} \in \boldsymbol{Q}} \frac{\sum_{k=1}^{n} P@k \cdot rel(k)}{R} \tag{2.7}$$

$$\boldsymbol{MRR} = \frac{1}{|\boldsymbol{Q}|} \sum_{\boldsymbol{q} \in \boldsymbol{Q}} \frac{1}{\text{rank of first relevant doc for } \boldsymbol{q}} \tag{2.8}$$

[6] See e.g. https://www-nlpir.nist.gov/projects/trecvid/trecvid.tools/trec_eval_video/A.README.

2.2. Learning to Rank

Both neural IR and LTR approaches can be formulated under the same, LTR, framework and have considerable similarities. We hence introduce the LTR paradigm in this section on the road to introducing neural IR approaches (which will be done in Section 2.3). We first discuss how LTR relate to the traditional ranking approaches (Section 2.2.1), then elaborate upon how LTR approaches are trained (Section 2.2.2).

2.2.1. From ranking to learning to rank

The previously discussed traditional (vector space and probabilistic) models employed an index of a document collection to rank documents in this collection and obtain a ranked result set per query. Learning to rank approaches can be used to subsequently re-rank the result set obtained with a traditional approach and can therefore be viewed as an *extension* of the original ranking process. LTR can thus be viewed as a two-step process (as displayed in Fig. 2.2):

Step 1 an *initial round of retrieval*, also known as **top-k retrieval**, in which a candidate set of (k) documents is obtained from the large document collection, for example using a simple but efficient traditional model (e.g. BM25);

Step 2 a *final round of retrieval*, in which the ranked result set of documents is obtained by feeding the candidate set and query to a computationally more expensive machine learning model to re-rank the documents, as displayed in the dashed box in Fig. 2.2.

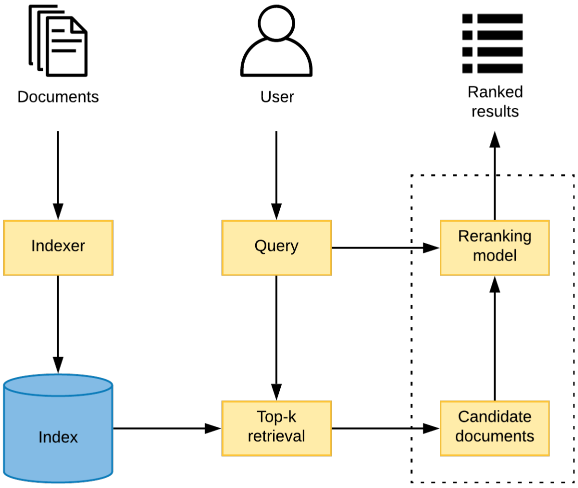


Figure 2.2: Graphical overview of the typical LTR process: different from the regular IR process, LTR approaches re-rank a subset of candidate documents.

Although most LTR systems follow the two-step process detailed above [53], we note that the candidate set of documents can also be obtained through other means, beyond the use of a different model for the initial round [12]. For example, it could be that a candidate set of documents is inherent in the dataset (e.g. all passages of Wikipedia page in a dataset that consists of questions on Wikipedia pages [22]).

2.2.2. Model learning

So far, we have assumed that an LTR approach has been trained, i.e. that the model has learned how to combine predefined features for ranking through its parameters that have been tuned. For the traditional models, we could tune their parameters by hand by testing a set of parameter configurations. For LTR approaches, on the other hand, different strategies for automatically tuning parameters have been proposed, as the complexity (i.e. the amount of possible ways to combine the predefined features) of these models may make it infeasible to tune the parameters by hand as we did for the traditional models [57]. These strategies, which can also be found in neural IR approaches, will be discussed in this section, after we first introduce a formal notation of LTR.

A formal notation of LTR

Recently, Guo et al. [49] have proposed a unified formal notation of document ranking based on the LTR framework. In this section, we will introduce a slightly modified version of their unified model formulation in the notation adopted throughout this thesis.

Suppose that $\boldsymbol{Q}$ is a set of queries and $\boldsymbol{D}$ is a set of documents in a corpus. Furthermore, suppose $\boldsymbol{Y}$ is a set of relevance labels, for which a total order exists (i.e. each relevance label in this set denotes a higher or lower relevance compared to every other relevance label in this set). Let $\boldsymbol{q_i} \in \boldsymbol{Q}$ be the i-th query (with $i \in \{1,\ldots,|\boldsymbol{Q}|\}$) and $\boldsymbol{D_i} = \{\boldsymbol{d_{i,1}}, \boldsymbol{d_{i,2}}, ..., \boldsymbol{d_{i,|D_i|}}\} \subseteq \boldsymbol{D}$ the set of documents associated with query $\boldsymbol{q_i}$. Moreover, let $\boldsymbol{Y}_i = \{y_{i,1}, y_{i,2}, ..., y_{i,|\boldsymbol{Y}_i|}\}$ be the relevance labels with respect to query $\boldsymbol{q_i}$ for each of the documents in the set $\boldsymbol{D_i}$, i.e. $y_{i,j}$ is the relevance label for document $\boldsymbol{d_{i,j}}$ with respect to query $\boldsymbol{q_i}$.

A ranking function $S(\boldsymbol{d_{i,j}}, \boldsymbol{q_i})$ then returns a relevance score for the query-document pair $\boldsymbol{q_i}, \boldsymbol{d_{i,j}}$. The general problem of document ranking is to minimize the difference between the relevance labels and the ranking resulting from the predicted relevance scores obtained with the scoring function S. In the learning to rank framework, this is expressed in a **loss function**. The loss function L can be expressed as the loss of retrieval function S over all queries $\boldsymbol{Q}$ and their associated documents out of $\boldsymbol{D}$, so the loss of retrieval function S for a document $\boldsymbol{d_{i,j}}$ with respect to query i is defined as $L(S; \boldsymbol{q_i}, \boldsymbol{d_{i,j}}, y_{i,j})$. In an LTR-formulation of a document ranking problem, we try to obtain the optimal ranking function $S^\star$ so that this loss is minimized, or formally:

$$S^\star = \arg\min \sum_i \sum_j L(S; \boldsymbol{d_{i,j}}, \boldsymbol{q_i}, y_{i,j}) \tag{2.9}$$

Learning objectives

Various learning objectives have been adopted for LTR models, three popular learning objectives are the pointwise, pairwise and listwise objectives. Hagen et al. [53] distinguishes them as follows:

- In the **pointwise approach**, machine learning methods are used for *each document* ($\boldsymbol{d_{i,j}}$) to predict the rank based on document-individual;
- In the **pairwise approach** *pairs of documents* ($\boldsymbol{d_{i,j}}, \boldsymbol{d_{i,k}}$) are used to conclude rank preferences for each pair;
- In the **listwise approach** a ranking function does not learn for individual documents or pairs, but processes *entire result lists* ($\boldsymbol{D_i}$).

In the experiments in this thesis, we adopt a pairwise training approach. A pairwise loss function can generally be formalized as [49]:

$$L(S; \boldsymbol{D}, \boldsymbol{Q}, \boldsymbol{Y}) = \sum_i \sum_{(j,k), y_{i,j} \succ y_{i,k}} L(S(\boldsymbol{d_{i,j}}, \boldsymbol{q_i}) - S(\boldsymbol{d_{i,k}}, \boldsymbol{q_i})) \tag{2.10}$$

where $\boldsymbol{d_{i,j}}$ and $\boldsymbol{d_{i,k}}$ are two documents for query $\boldsymbol{q_i}$ and $\boldsymbol{d_{i,j}}$ is preferred over $\boldsymbol{d_{i,k}}$ (i.e. $y_{i,j} \succ y_{i,k}$). More specifically, we employ the well-known **hinge loss** function in our experiments, which is defined as [49]:

$$L(S; \boldsymbol{D}, \boldsymbol{Q}, \boldsymbol{Y}) = \sum_i \sum_{(j,k), y_{i,j} \succ y_{i,k}} \max(0, 1 - (S(\boldsymbol{d_{i,j}}, \boldsymbol{q_i}) - S(\boldsymbol{d_{i,k}}, \boldsymbol{q_i}))) \tag{2.11}$$

As can be seen, the hinge loss not only requires a model to score $\boldsymbol{d_{i,j}}$ higher than $\boldsymbol{d_{i,k}}$ for $\boldsymbol{q_i}$, but also penalizes a retrieval function S if the difference between both scores (the *margin*) is smaller than 1.

2.3. Neural IR

Whereas LTR methods employ handcrafted features to represent input, neural IR models automatically learn which features should be employed to represent input. Neural IR approaches do so by employing deep learning techniques, which are also called representation learning techniques [71]. As a result of these techniques, deep learning approaches are capable of detecting hidden and complex data patterns, that are difficult to capture with shallow neural networks or approaches based on hand-crafted features, that are thereby sometimes outperformed [34]. In the remainder of this section, we provide a holistic view of neural IR models (Section 2.3.1), followed by various categorizations of neural IR models (Section 2.3.2) and finally introduce the neural IR models studied in this thesis (Section 2.3.3)

2.3.1. A holistic view

Document ranking can be defined as a matching problem, that consist of three primary steps: generating a representation of the query, generating a representation of the considered document and matching both representations to estimate the relevance of the document to the query [86]. As displayed in Fig. 2.3, neural models can be used to generate both representations, but also to estimate relevance.

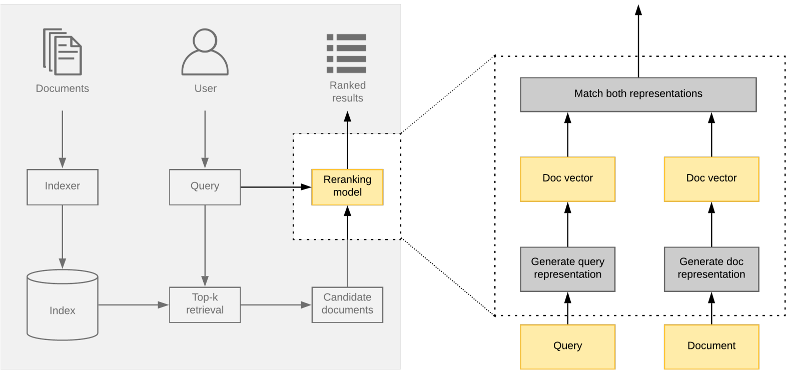


Figure 2.3: Graphical overview of neural models (right in color) within the LTR framework (left in grayscale), inspired by [85]: neural models can be useful either for generating good representations or in matching a query and a document representation, or both.

We can include such components in our formal notation of the ranking function S introduced in Section 2.2.2, by adopting the formulation proposed in [49]:

$$S(\boldsymbol{d},\boldsymbol{q}) = g(\psi(\boldsymbol{q}),\phi(\boldsymbol{d}),\eta(\boldsymbol{d},\boldsymbol{q})) \tag{2.12}$$

in which:

- $\boldsymbol{q}$ and $\boldsymbol{d}$ are respectively the input query and document;
- ψ and ϕ are the respective **representation functions** that extract features from $\boldsymbol{q}$ and $\boldsymbol{d}$;
- η is the **interaction function** that extracts features from the query-document pair $(\boldsymbol{q},\boldsymbol{d})$;
- g is the **evaluation function** which computes the relevance score based on the feature representations.

Note that ψ and ϕ are evidently found in the bottom two rectangular boxes in Fig. 2.3, whereas η and g—which require input on the query *and* document—both preside in the top rectangular box in Fig. 2.3.

This formulation allows us to clearly distinguish learning to rank approaches from neural IR approaches. Traditional LTR approaches adopt fixed functions for ψ,ϕ and η (i.e. manually defined feature functions) and a machine learning model for g (e.g. logistic regression), whereas neural IR approaches typically encode these functions in network structures so that all of them can be learned from the data. Moreover, the inputs $\boldsymbol{q}$ and $\boldsymbol{d}$ are in the LTR case usually raw texts, whereas in neural approaches these inputs could either be raw texts or word embeddings. Such **embeddings** can be used to obtain vector representations of words and are often pre-trained in an unsupervised manner [86]. Many neural approaches, including the approaches studied in this work, employ word embeddings. However, embedding mapping is not considered as part of ψ,ϕ or η, but as a basic input layer. A reason to exclude this embedding process from the representation and interaction functions is that different embeddings can be employed with the *same* neural model. Hence, as we want to introduce the differences between models, we exclude this component from our comparison and focus on the neural ranking models that may use such embeddings and are trained in a supervised fashion.

2.3.2. Categories of neural IR models

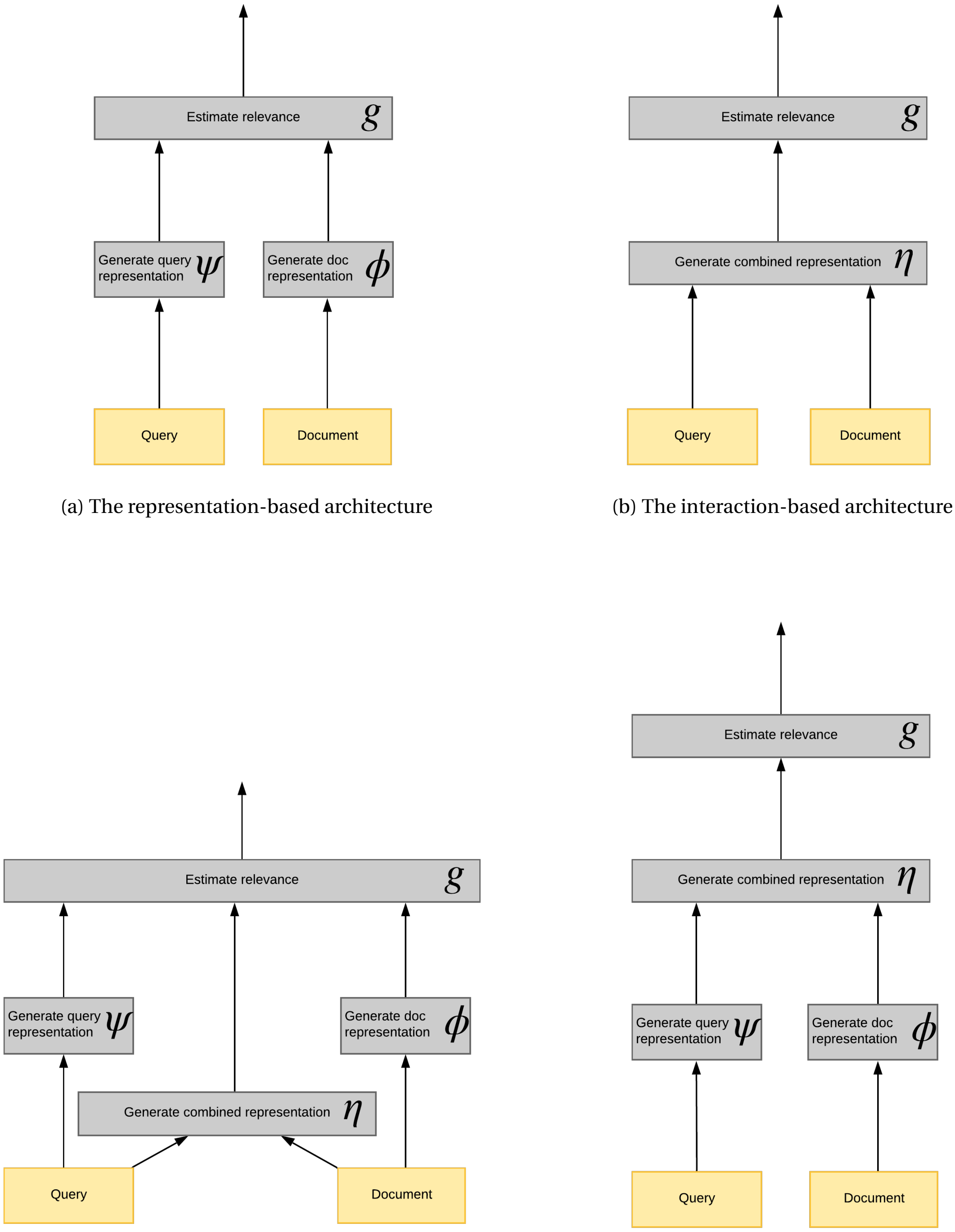
The large number of proposed neural models for IR can be categorized in various ways, which have been enumerated by [49] and will be detailed in this section. The most widely used categorization (in e.g. [48, 49, 87, 97, 131]) distinguishes **interaction-based, representation-based and hybrid approaches**, based on the manner they model the query and document. Representation-based approaches strive to create good representations of the query and the document through ψ and ϕ respectively, which both typically consist of several hidden layers. Representation-based models ultimately combine the output signals of ψ and ϕ by applying a simple similarity function on the last level. On the other hand, interaction-based approaches directly model the local interactions between the query and document through an interaction matrix (η). Afterwards, interaction-based approaches fed this matching matrix into a deep neural network to obtain the document relevance score [120]. Hybrid approaches incorporate both interaction- and representation-based input and employ ψ, ϕ *and* η, either separately [87] or subsequently [126] before obtaining the relevance score with g. Figure 2.4 displays a graphical representation of the three categories of neural IR models.

Since representation-based approaches evaluate relevance based on high-level (semantic) representations of inputs, they better fit short input texts (for which it is easier to obtain good high-level representations compared to long texts). Interaction-based approaches on the other hand, employ detailed interaction signals and better fit tasks that ask for specific matching patterns such as exact word matching. They are also considered more suitable for processing heterogeneous inputs (i.e. documents that are much longer than queries), as they avoid the issue of encoding long texts.

Despite the clear motivation for representation-based approaches and the need for semantics over syntax matching, a recent comparative study by Nie et al. [93] has shown the deep interaction-based approaches to clearly outperform the representation-based approaches in terms of retrieval effectiveness, albeit at the cost of some efficiency [83]. Moreover, Pang et al. [98] states that interaction-based approaches are more popular because they stand closer to the also popular (original) LTR methodology and benefit from the interaction-matrix's ability to reveal relevance signals in a visual manner.

Another categorization, used in e.g. Guo et al. [49], distinguishes **symmetric and asymmetric approaches**. *Symmetric approaches* assume inputs $\boldsymbol{q}$ and $\boldsymbol{d}$ to be homogeneous, i.e. $\boldsymbol{q}$ and $\boldsymbol{d}$ can be interchanged in the input layer, without affecting the final output. Hence, models that fall under this category take two "sentences" or "texts" as input, rather than a discriminative query and document. The representation-based approaches that are symmetric are also known as *siamese networks* [136]: in these networks there exist no separate ψ and ϕ, or if you want, $\psi = \phi$. On the other hand, interaction-based approaches that are symmetric employ components that are symmetric by definition for η, such as operations on pairs of n-grams of terms in $\boldsymbol{q}$ and $\boldsymbol{d}$ [58]. The other category (of *asymmetric approaches*) assume inputs $\boldsymbol{q}$ and $\boldsymbol{d}$ to be heterogeneous, i.e. inputs $\boldsymbol{q}$ and $\boldsymbol{d}$ follow a different path within the deep network until they are finally combined to estimate relevance in the final layer. Such architectures have mainly been introduced for the ad-hoc task, due to the inherent heterogeneity between the query and document as discussed in Section 2.1.1.

Finally, a third categorization distinguishes **single-granularity and multi-granularity approaches** [49]. In single-granularity approaches, g only takes the final outputs of functions ψ and ϕ and/or η for relevance computations. In multi-granularity approaches, g employs intermediary outputs of functions ψ and ϕ so that it can estimate relevance based upon multiple granularities. Hence, multi-granularity approaches can benefit from using different levels of feature extractions (as is done in e.g. [137]) or different units of language (e.g. words, phrases *and* sentences) as is done in [31]. However, all models covered in the experiments conducted in this thesis are single-granularity models.


Estimate relevance g
Generate query representation ψ
Generate doc representation ϕ
Query
Document
Estimate relevance g
Generate combined representation η
Query
Document
Estimate relevance g
Generate query representation ψ
Generate doc representation ϕ
Generate combined representation η
Query
Document
Estimate relevance g
Generate combined representation η
Generate query representation ψ
Generate doc representation ϕ
Query
Document

(a) The representation-based architecture

(b) The interaction-based architecture

(c) The hybrid Duet architecture [87]

(d) The hybrid MV-LSTM architecture [126]

Figure 2.4: Abstract overviews of the architecture of neural IR models, which can be categorized on the manner they model the query and document: focusing on representations (displayed in 2.4a), interactions (displayed in 2.4b) or both (displayed in 2.4c,2.4d). The latter—hybrid—approach can take various forms of which we here display two.

2.3.3. Covered neural models

We will now discuss each of the six neural IR models studied in this work. For each model we will elaborate upon the intuition behind it and classify it according to the introduced categorizations in Section 2.3.2.

ARC-I [58] (ARChitecture-I), displayed in Fig. 2.5, was proposed in 2014 among the earlier neural IR models. In essence, it considers a matching algorithm that adopts a convolutional strategy (proven successful in computer vision image and speech recognition) to obtain representations of natural language. ARC-I is a symmetric, representation-based approach, i.e. a siamese network. It separately summarizes the meaning of two sentences (or more concretely, the embedding of words in both sentences) through one dimensional (1D) layers of convolution and pooling (ϕ) and finally compares the representation of two sentences with a multilayer perceptron (MLP) (g). The model evidently suffers from a drawback inherited from the siamese architecture: it defers the interaction between two sentences to until their individual representation matures (in the convolutional model), and runs at the risk of losing details (e.g. a city name) important for the matching task at hand.

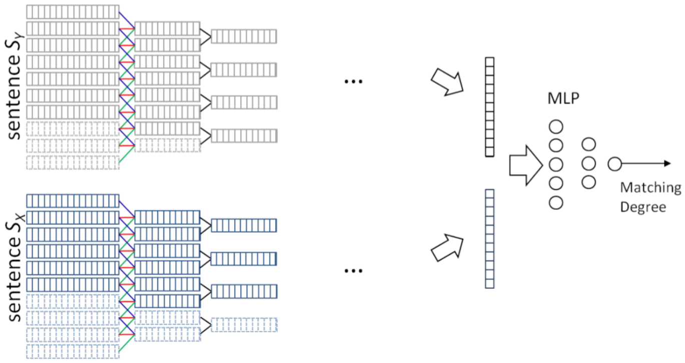


Figure 2.5: Overview of the ARC-I architecture, copied from [58].

MatchPyramid [96], displayed in Fig. 2.6, also adopts a convolutional strategy to obtain representations of natural language in a symmetric manner. However, different from ARC-I, MatchPyramid is an interaction-based approach: instead of first obtaining separate representations of two sentences, MatchPyramid directly employs a 2D matching matrix to capture interactions through the dot product of embedded words originating from the sentences (η). This approach was motivated by the intuition that a good matching model should account for matching various patterns beyond exact positional matches. Subsequently, similar to ARC-I, the model employs several convolutional neural networks (CNNs) and pooling, although in a two-dimensional manner instead of one-dimensional manner, and finally an MLP to compute the matching score (g). The name “MatchPyramid” stems from the matching problem for which is was created and the matching matrix (i.e. the bottom of a pyramid) that is transformed into smaller and smaller 2D layers that ultimately provide a single matching scores (i.e. the top of a pyramid).

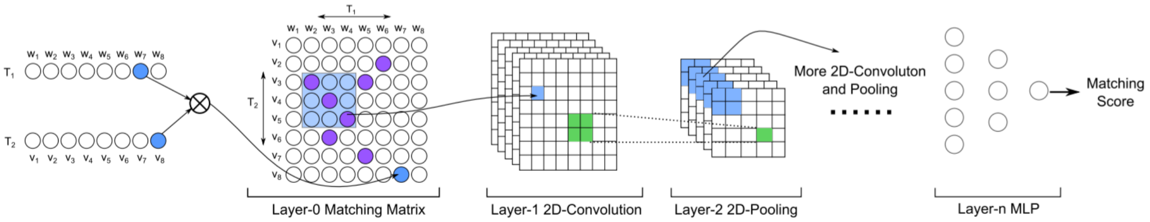

T_1
w_1 w_2 w_3 w_4 w_5 w_6 w_7 w_8
T_2
v_1 v_2 v_3 v_4 v_5 v_6 v_7 v_8
More 2D-Convoluton and Pooling
......
Matching Score
Layer-0 Matching Matrix
Layer-1 2D-Convolution
Layer-2 2D-Pooling
Layer-n MLP

Figure 2.6: Overview of the MatchPyramid architecture, copied from [97].

MV-LSTM [126] (Multiple-Views Long Short-Term Memory), displayed in Fig. 2.7, is another symmetric approach, that aims to capture a representation of context rather than separate words, by employing multiple representations (views) of sentences that each focus on different parts of local information. Some consider MV-LSTM to be a representation-based approach (e.g. in [49]), but we here view it as a hybrid approach as it both obtains a separate representation of two sentences and then captures their interactions with an interaction matrix, before transforming the representations into a final matching score. More specifically, MV-LSTM first generates *positional representations* for each sentence using a bi-directional LSTM ($\phi = \psi$), i.e. it concatenates a representation of the whole sentence from a forward and backward direction up to the position of a word (displayed as a dashed orange box in Fig. 2.7: the orange-filled boxes represent the forward and backward representation for one word in S_x). Subsequently, an interaction tensor is employed to model interactions between any two positional representations in a two-dimensional manner (η), after which, k-max pooling and an MLP are used to obtain a final score (g). We however do not view MV-LSTM as a multi-granularity approach (following [49]), as it does not consider multiple granularities (e.g. characters/words/sentences) but multiple positions (e.g. all words before and after a word).

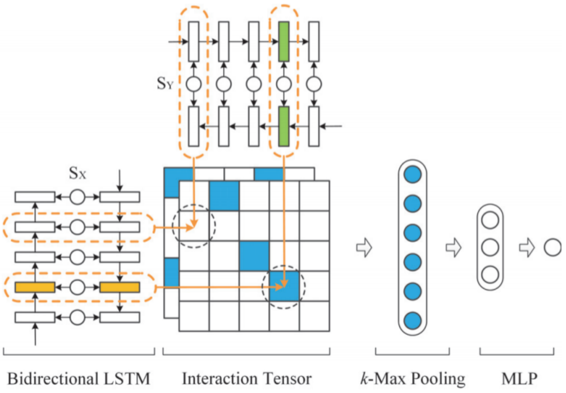


Figure 2.7: Overview of the MV-LSTM architecture, copied from [126].

Duet, displayed in Fig. 2.8, is an asymmetric, hybrid approach. It owes its name to employing two subnetworks in parallel: a lexical matching or local model and a semantic matching or distributed model. The lexical matching sub-network first fills a 2D binary matching matrix with exact positional matches of words of a query and a document (η) and then employs a 2D CNN and several fully connected-layers to obtain a local score (as part of g). The semantic matching sub-network learns representations of query (ϕ) and document (ψ) and then computes the positional similarity of query and document terms using n-graphs, followed by a matching through the Hadamard product and fully connected-layers to obtain a distributed score (as part of g). Finally, the sum is taken over the output of both sub-networks (as the final part of g).

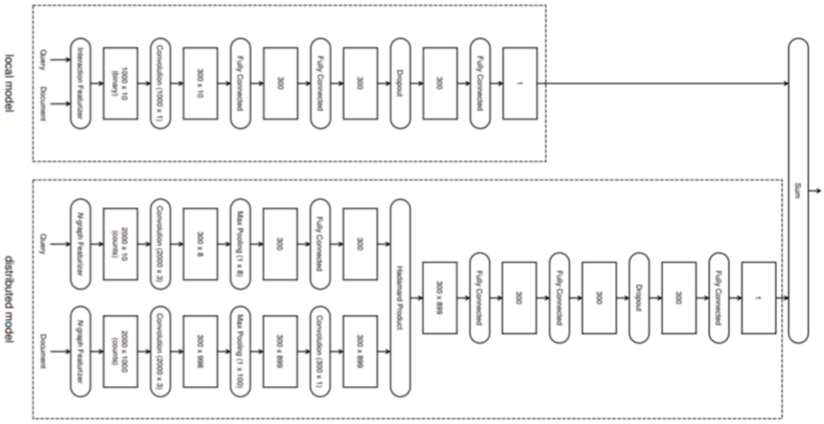


Figure 2.8: Overview of the Duet architecture, copied from [87].

DRMM [48], displayed in Fig. 2.9, an interaction-based model that employs a matching histogram mapping of the similarity between each query term and the document (η). Specifically, DRMM first obtains similarity scores between a query term and document terms with a cosine similarity function. Subsequently, it uses these similarity scores to fill fixed-length matching histograms (by discretizing the similarity scores into equal-sized bins and assigning the resulting scores to corresponding bins [13]). Subsequently, DRMM feeds each query term matching histogram to a feed-forward network to compute relevance at the term-level and employs a term gating network (i.e. an aggregation based on query term importance) before aggregating the obtained scores into a final score (g).

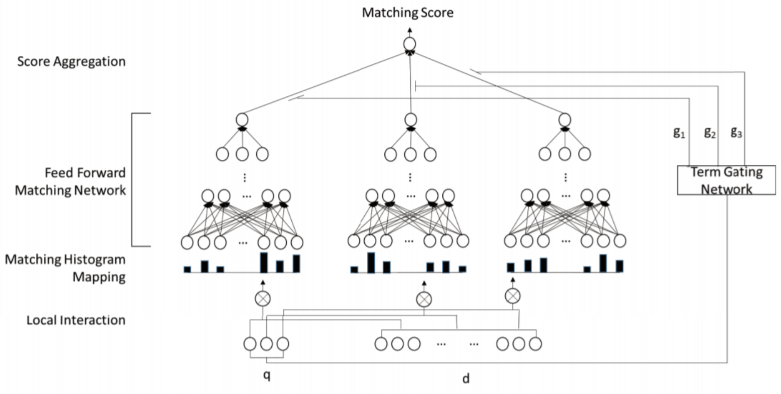


Figure 2.9: Overview of the DRMM architecture, copied from [48].

aNMM [133] (attention-based Neural Matching Model) was specifically designed for ranking short text in an interaction-based fashion. Yang et al. [133] actually proposed two aNMM architectures: aNMM-1 and aNMM-2, of which we have employed aNMM-1 to which we will simply refer as aNMM. Similar as DRMM, aNMM consists of three steps: a matching matrix (η), a deep neural network with value-shared weighting scheme in the first layer and fully connected layers in the rest (as part of g) and finally a question attention network to learn question term importance and produce a final score (final part of g). The value-shared weighting scheme, different from the position-shared weighting employed in e.g. MatchPyramid, was designed to capture the importance of different levels of (semantic) matching signals. The question attention network is similar to the query term gating network in DRMM. Since only an overview of the aNMM-2 architecture is made available, we present an overview of this architecture in Fig. 2.10. Different from the displayed figure of aNMM-2, aNMM-1 only adopts a single set of value-shared weights (i.e. instead of weights per node as displayed for the yellow nodes in the figure, we only have *one* set of weights).

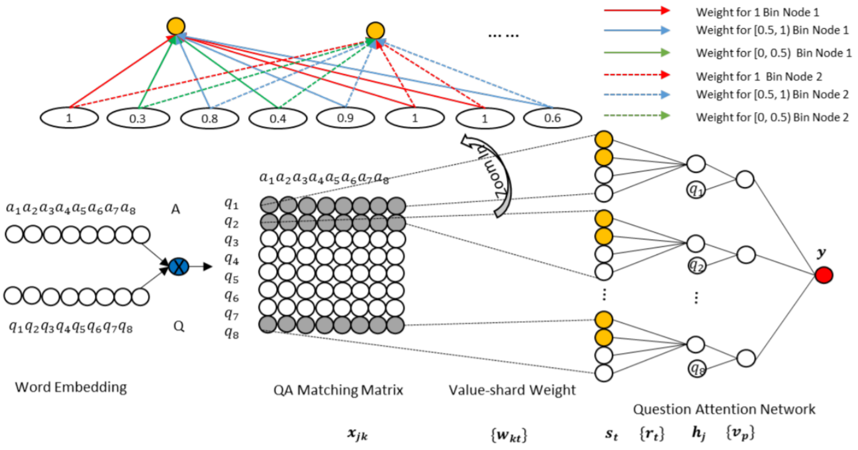


Figure 2.10: Overview of the aNMM-**2** architecture, copied from [133].

2.4. Evaluating Deep Learning Approaches

In the following, we will first discuss how deep learning models have been evaluated in IR in Section 2.4.1 and then introduce means for evaluation as proposed in related fields beyond IR in Section 2.4.2.

2.4.1. Evaluations in neural IR

Today's de facto methodology for evaluation is to test a model on benchmark collections, following the traditional Cranfield Paradigm, introduced in Section 2.1.3. In the following we will first introduce shortcomings in this approach as employed in neural IR research. Subsequently, we will discuss works that have tried to go from a focus on evaluation (knowing which model is better) to analysis (understanding why a model works better).

Problems in widely adopted evaluation methods

Among benchmark collections employed in neural IR, we can find datasets from the TREC Robust [48, 83, 138] Web [59, 60] and QA [133] tracks, but also e.g. a Yahoo! QA dataset [126, 127]—for which all relevance judgments are publicly available. Evaluations on these collections can show us, for a given dataset, whether a model (significantly) outperforms another model. A separate error analysis is then required to try to identify where a performance increase (or loss) comes from.

However, many works that have employed such benchmarks, justify their model by outperforming other neural approaches and decades-old baselines such as BM25 [48, 59, 60, 138], instead of typically stronger baselines such as RM3 or Rocchio's classifier or the best known performance on a specific task (as is only done by a few works, such as [120] and [47]). For example, a state-of-the-art neural model achieves a MAP of 0.28 on the Robust 2004 dataset and therefore beats both BM25 and QL as well as other neural approaches [48], but the best submission to the original track already achieved a MAP of 0.32 [125]. Moreover, [60] and [59] proposed neural models (resp., CO-PACRR and PACRR - in the IR community considered to be state-of-the-art models) and report positive improvements on ERR@20[7] across the TREC Web Tracks 2010-14 compared to other state-of-the-art neural approaches. However, if we compare their best runs in to the best submissions to the original track [16, 24, 25, 27, 116], we find that they only achieve gains on 3/6 years and even losses on 2/6 years. These skewed images of retrieval effectiveness are neither limited to these examples [73] nor limited to neural approaches proposed over the last decade [7]. Hence, we conclude that benchmark evaluations against weak baselines can give a too positive image of retrieval models [7, 73].

Year	Best ERR@20	(CO-)PACRR ERR@20	Increase of (Co-)PACRR over best
2010	0.166	0.160	-0.006
2011	0.157	0.167	+0.010
2012	0.313	0.363	+0.050
2013	0.184	0.189	+0.005
2014	0.233	0.232	-0.001

Table 2.1: Difference in performance between the best known performance [16, 24, 25, 27, 116] and the best performance obtained with (Co-)PACRR [59, 60] as measured in ERR@20 per year for the TREC Web tracks 2010-2014.

Recently, the IR community has seen the rise of some leaderboards with held-out relevance judgments, such as `MSMarco` [92]. Evidently, such leaderboards do not allow to justify models by outperforming weak baselines. However, they suffer from the lack of testing statistical significance and do not allow researchers to conduct error analyses. Specifically with regard to deep learning approaches which have an opaque nature, it is difficult to comprehend their inner workings and confidently point to sources for performance deviations. Deviations may possibly come from the intuition researchers have proposed with their model, but with leaderboards there is no way to validate this with an error analysis. We conclude that this standard may give rise to a disproportional focus on maximizing quantitative improvements, while neglecting theoretical understanding and qualitative insights in the process, as expressed by Mitra and Craswell [85].

From evaluation towards analysis

Over the past year, some methods to move beyond Cranfield experiments for analyzing neural approaches, have been proposed. Nie et al. [93] conducted a comparison of representation- and interaction-based IR

[7] ERR@20 considers the *Expected Reciprocal Rank* metric designed for multi-graded relevance. It specifically models how long it will take for a user to find a relevant document [14].

models (which typically have been trained and tested on different datasets), on the same training and testing collection. While such work enables us to empirically determine which type of approach performs better, they can only provide insights at a level of high abstractness (that was criticized by Fang et al. [42], as we introduced in Section 1.2.1). For example, knowing that interaction-based models outperform representation-based models [93] does neither shed much light on *why* this is the case nor whether *combining* models would make sense. Along a different line, Cohen et al. [21] recently proposed to *probe* neural retrieval models by training them, and then using each layer's weights as input to a classifier for different types of NLP tasks (sentiment analysis, part-of-speech tagging, etc.). The motivation behind this approach is that the performance on those tasks by each network layer provides insights into the kind of information that each layer captures. While this is indeed useful to realize, it does not provide an immediate insight into how to improve an existing neural approach, as we do not how the NLP task relates to a considered IR task.Moreover, the approach is not model-agnostic and hence requires a different implementation per model. In a similar fashion, some works like [60, 83] have conducted an ablation study in which the influence of various model components was researched. Although this can provide a reason to include or exclude components (e.g. context-sensitive term encoding [83] and a disambiguation, a cascade k-max pooling or a shuffling combination layer [60]), it does not shed much light on *why* inclusion or exclusion of a component improves retrieval effectiveness.

The mainstream approach for evaluating neural IR models suffers from a limited scope of evaluation and proposed solutions offer little insight and/or are labor-intensive since they require a different implementation per model. In contrast, we propose a model-agnostic method that is less restricted in scope compared to existing solutions, as its scope is determined by the included axioms that cover many IR components (such as term frequency, inverse document frequency, length normalization, semantics, regularization, proximity).

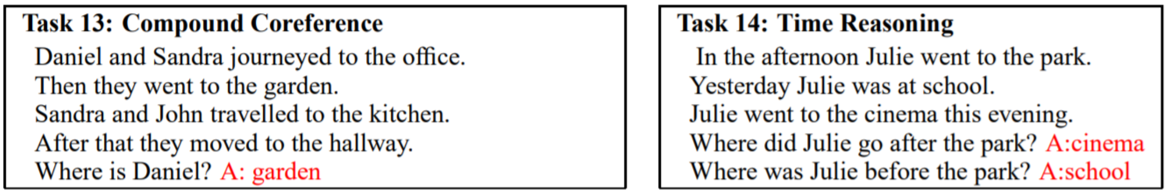
2.4.2. Evaluation in other research fields

Looking at the NLP community that is strongly related to our IR community, we can obtain that it faces hundreds of works that vy for leaderboard dominance while basic questions remain unanswered [68, 73]. Among the few works that have addressed basic questions, we can find the work of Kaushik and Lipton [68], which addresses the question of identifying the difficulty of several popular reading comprehension benchmarks in NLP. They, identified surprising unwanted characteristics in some datasets. For example, they found that the `Children's Books Test` (CBT) can be gamed. CBT was designed to capture how well language models, that get a passage from a children's book, a question and a set of candidate answers as input, are able to capture the meaning of the passage so that they can answer the question [56]. Kaushik and Lipton [68] found that the test could be gamed (a performance increase could be obtained) by only looking at the last sentence. Such findings justify the critique on solely employing benchmarks for evaluating a new model, introduced in Section 1.2.

However, whereas evaluation methodologies of deep nets beyond benchmark datasets are in their infancy in the IR community, the NLP and Computer Vision communities have proposed a number of them. Weston et al. created a set of 20 so-called `bAbI` tasks: each task consisting of several QA instances, aimed at diagnosing some form of text understanding and reasoning [129]. Two examples of such tasks are displayed in Fig. 2.11. Within the field of Visual Question Answering (VQA), Johnson et al. [66] developed `CLEVR`, a dataset for language and visual reasoning, consisting of a large number of rendered images (constructed from a limited universe of objects and relationships) and automatically generated questions [66], as displayed in Fig. 2.12. Along a different line, Jia and Liang [65] proposed an adversarial evaluation scheme of the `SQUAD` dataset by inserting distracting sentences into text passages, resulting in a sharp drop in accuracy across all evaluated models. An example of

Although they can provide more insights into why and when certain approaches work better than others, the proposed solutions also have shortcomings. For example, the `bAbI` tasks have been criticized, since some of the tasks can be solved (almost) as good by only looking at the passages (and not at the question!), which is also the case for Task 13 displayed in Fig. 2.11 [68]. Next to this error in the construction of the diagnostic instances, a shortcoming of the listed diagnostic approaches is that they have come up with their own features for diagnosis. In the publications introducing `bAbI` and `CLEVR`, the authors do not state why the respective 20 tasks and 90 question templates were selected.

In spite of these shortcomings, we propose to bring the approach of diagnostic dataset creation into the neural IR community with this thesis. Different from the proposed solutions, we base our diagnostics upon well-established heuristics in the field of IR, known as axioms. Before discussing the creation of diagnostic datasets in Chapter 4, we will first introduce the paradigm of axiomatic thinking in IR.



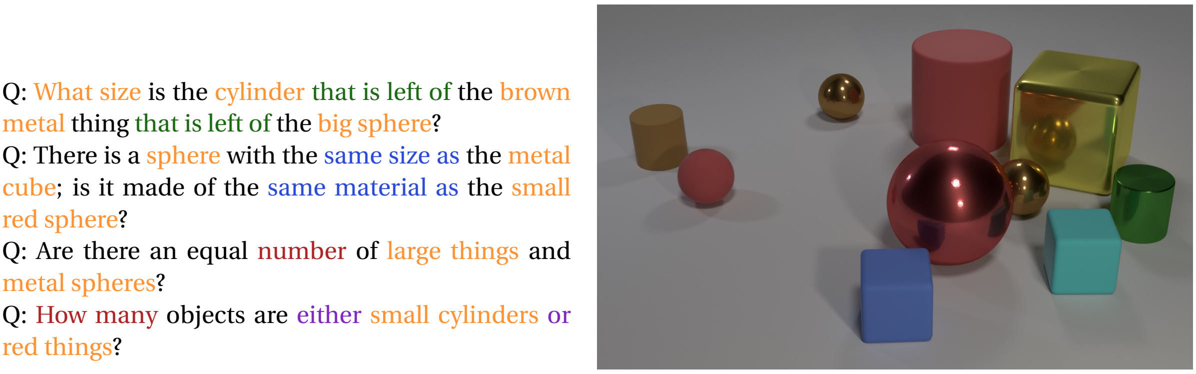
Task 13: Compound Coreference

Daniel and Sandra journeyed to the office.
Then they went to the garden.
Sandra and John travelled to the kitchen.
After that they moved to the hallway.
Where is Daniel? A: garden

Task 14: Time Reasoning

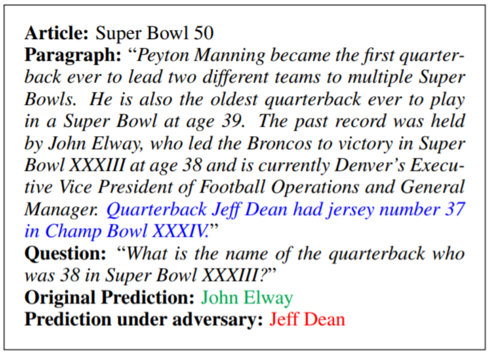
In the afternoon Julie went to the park.
Yesterday Julie was at school.
Julie went to the cinema this evening.
Where did Julie go after the park? A:cinema
Where was Julie before the park? A:school

Figure 2.11: Examples for two of the `bAbI` tasks, adapted from [129]. Task 13 tests coreference in the case where the pronoun can refer to multiple actors, task 14 tests the understanding of time expressions within statements.



Q: What size is the cylinder that is left of the brown metal thing that is left of the big sphere?
Q: There is a sphere with the same size as the metal cube; is it made of the same material as the small red sphere?
Q: Are there an equal number of large things and metal spheres?
Q: How many objects are either small cylinders or red things?

Figure 2.12: A sample image and associated questions from the `CLEVR` dataset, adapted from [66]. The questions test aspects of visual reasoning, such as attribute identification, counting, comparison, multiple attention, and logical operations. Both the images and questions were automatically created based upon question templates and input on e.g. the universe of objects that can be present in images. Hence, there is no shortage of diagnostic data per capability to diagnose.



Article: Super Bowl 50
Paragraph: "*Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager. Quarterback Jeff Dean had jersey number 37 in Champ Bowl XXXIV.*"
Question: "*What is the name of the quarterback who was 38 in Super Bowl XXXIII?*"
Original Prediction: John Elway
Prediction under adversary: Jeff Dean

Figure 2.13: Example from the `SQuAD` dataset [102] including an adverserial distracting sentence added in [65]. A BiDAF ensemble model originally returns the correct answer, but is fooled by the adverserial approach.

2.5. Axiomatic Thinking in IR

In this section we introduce the paradigm of axiomatic thinking in IR. Starting with a paper by Fang et al. [42] considering a manual analysis of six axioms on two existing retrieval functions—discussed in Section 2.5.1—subsequent works on axiomatic thinking have considered more axioms and another diagnosis strategy and have resulted in novel retrieval functions and metrics and a novel re-ranking approach—discussed in Section 2.5.2.

2.5.1. A manual analysis of six axioms

Hui Fang, Tao Tao and ChengXiang Zhai can be considered the founding fathers for the adoption of axiomatic thinking in IR. In their seminal work, [42], they introduced six retrieval constraints that any reasonable retrieval function should satisfy. Formalizing retrieval heuristics into constraintsenabled the authors to *analytically* evaluate a number of existing retrieval functions. This analytical approach consisted of looking at a retrieval formula and an axiom to conclude under which conditions (if any) the retrieval formula would fulfill the axiom.

For example, we can obtain that Okapi BM25 (displayed in Eq. 2.13), in case query $\boldsymbol{q} = \{w\}$, will assign a higher score to a document that contains more occurrences of the query term (i.e. has a larger $c(w, \boldsymbol{d})$) under the condition that the query term w occurs in no more than half of the documents ($df(w) \le N/2$): we have to avoid the case in which a query the highlighted part of the BM25 retrieval formula becomes negative and hence assigns a lower score to a document that has a higher count of the query term. From this, Fang et al. [42] could conclude that Okapi BM25 fulfills TFC1, displayed in Eq. 2.14, under the condition that $df(w) \le N/2$.

$$BM25(\boldsymbol{d}, \boldsymbol{q}) = \sum_{w \in \boldsymbol{q} \cap \boldsymbol{d}} \left(\ln \frac{N - df(w) + 0.5}{df(w) + 0.5} \times \frac{(k_1 + 1)c(w, \boldsymbol{d})}{k_1(1 - b + b\frac{|\boldsymbol{d}|}{avdl}) + c(w, \boldsymbol{d})} \times \frac{(k_3 + 1) \times c(w, \boldsymbol{q})}{k_3 + c(w, \boldsymbol{q})} \right) \tag{2.13}$$

Equation 2.13: Retrieval formula of Okapi BM25. The IDF part of the formula is highlighted.

$$\text{Let } \boldsymbol{q} = \{w\} \text{ and assume } |\boldsymbol{d_1}| = |\boldsymbol{d_2}|. \text{ If } c(w, \boldsymbol{d_1}) > c(w, \boldsymbol{d_2}), \text{ then } S(\boldsymbol{d_1}, \boldsymbol{q}) > S(\boldsymbol{d_2}, \boldsymbol{q}). \tag{2.14}$$

Equation 2.14: A formal expression of the TFC1 axiom [42]. Okapi BM25 does not fulfill this axiom if $df(w) > N/2$.

The main assumption of this approach—retrieval effectiveness is closely related to the fulfillment of retrieval constraints—was empirically validated. Fang et al. [42] adapted the retrieval formula of Okapi BM25 so that it fulfilled the TFC1 constraint (among other constraints) and obtained an increase in average precision on various TREC test collections (0.04-0.27 for verbose and 0-0.01 for non-verbose queries) with the modified version of BM25, displayed in Eq. 2.15.

$$BM25(\boldsymbol{d}, \boldsymbol{q}) = \sum_{w \in \boldsymbol{q} \cap \boldsymbol{d}} \left(\ln \frac{N + 1}{df(w)} \times \frac{(k_1 + 1)c(w, \boldsymbol{d})}{k_1(1 - b + b\frac{|\boldsymbol{d}|}{avdl}) + c(w, \boldsymbol{d})} \times \frac{(k_3 + 1) \times c(w, \boldsymbol{q})}{k_3 + c(w, \boldsymbol{q})} \right) \tag{2.15}$$

Equation 2.5: Adapted version of the retrieval formula of Okapi BM25: the highlighted IDF part of the equation can no longer be negative and hence this formula fulfills TFC1 unconditionally.

2.5.2. Rise of the axiomatic thinking paradigm

Following the seminal work by Fang et al. [42], various works have contributed to the field of axiomatic thinking, which we can by now define as a paradigm on its own. In the following we will subsequently discuss various developments within this paradigm.

Novel retrieval functions and more axioms

In follow-up works, Fang and Zhai [40] also derived novel retrieval functions such as F1-LOG and F2-EXP, based on their initial set of constraints. Later, they extended their list of axioms from exact (syntactical) term-matching to semantic-matching based constraints [39, 41]. Others have contributed query term proximity [53, 121], document length normalization [78] and query term discrimination [6] constraints, again consistently showing that traditional retrieval models improve when slightly altered to satisfy those constraints.

While most of the more than twenty existing axioms have been designed for standard retrieval models, a number of axioms have also been proposed for the more specialized cases, such as statistical translation models [67] and pseudo-relevance feedback [18, 19, 89].

Diagnosis through collection perturbations

Apart from the manual inspection of axiom fulfillment, discussed in Section 2.5.1, Fang et al. [43] also researched a second strategy: the use of collection perturbations. The original documents in the document collection were thereby adapted with relevance preserving perturbations (their relevance labels would remain the same and so would the queries). With these perturbations, Fang et al. [43] created documents for a specific test scenario, which could be related to a specific axiom. For example, length variance amplification was realized by appending documents to themselves (linear to their original length, so larger documents grow faster) to test whether a model is robust to larger document length variance (related to the document length normalization constraint, which will be discussed in Section 3.3). These tests provided findings consistent with the first strategy of analytical validation, but can also be used if manual analysis is challenging and provide further insights on retrieval functions fulfilling the same set of axioms.

Nevertheless, this strategy has, different from the first, not been employed in subsequent works other than [90]. One reason for this may be that the first approach could *predict* the performance of a retrieval function (a longstanding challenge in IR [30]), whereas the second approach requires additional experiments.

Diagnosing evaluation metrics

Next to the line of research that focuses on the diagnosis of models, the axiomatic thinking paradigm has been used to diagnose evaluation metrics. In general, constraints or axioms were used to formally specify how metrics should behave in particular situations [5]. For example, the priority constraint states that swapping items in concordance with their relevance should increase the ranking quality score [5]. Such formalized heuristics again allowed researchers to identify strengths and weaknesses but now in (widely adopted) metrics [88, 117] and resulted in the proposal of new metrics such as reliability and sensitivity [4].

Towards multi-term queries

Lastly, we point to a work closest to ours. Hagen et al. [53] explored the re-ranking of a given result list based on the aggregated re-ranking preferences of twenty-three axioms. They found that axiom-based re-ranking could improve retrieval performance for almost all 16 basic retrieval models. Similar to our work, this application of axioms to an actual result list (instead of artificial documents with one or two terms as in the analytic evaluation of retrieval functions) requires the *extension* and *relaxation* of axioms. On a higher level, our work is different to all those introduced above in the sense that we create *datasets* (one per axiom), in order to determine empirically to what extent neural IR approaches satisfy the individual axiomatic constraints.

Hagen et al. [53] could not include the document length axiom $QNLC$ and $STMC3$ constraints proposed by Fang and Zhai [40, 41] and proposed six new proximity axioms. Our axiom conversion (extension and relaxation) scheme does not have issue with the aforementioned constraints and can be used to convert an axiom and subsequently create a diagnostic dataset (through artificial data creation, if needed) for any axiom on retrieval status scores if it can be expressed in measurable information retrieval statistics (including the newly proposed axioms). For example, the proximity constraints proposed by Fang and Zhai [40] prescribe a relation on measures of proximity rather than retrieval status values and can not be used to diagnose IR models with our approach. Similarly the term semantic similarity constraints proposed by Fang and Zhai [41] prescribe a similarity between a query and terms and can also not be used in our method.

Along a different line, focusing on augmenting the training regime of neural models with axiomatic knowledge, Rosset et al. [108] have recently considered the direct incorporation of axioms in the loss function. After training a deep net using this loss function, they achieved significant improvements in retrieval effectiveness over a default training regime. To obtain instances that match the conditions of axioms, they propose to perturb documents in the document collection to make them fulfill the conditions of an axiom and subsequently use a pair of a regular and a perturbed version of a documents (that together fulfill the conditions of an axiom) to augment the training scheme of a deep net.

3

Creating Diagnostic Datasets

In Section 1.2 we argued our choice for diagnosing neural models through an axiomatic approach. In this chapter we discuss how we have created the resources for such a diagnosis: the diagnostic datasets. First, in Section 3.1, we introduce the motivation for creating diagnostic dataset, followed by the proposed methodology for creating diagnostic instances in Section 3.2. This methodology consists of two steps: 1) axiom conversion and 2) diagnostic dataset creation. We have executed this methodology on four established axioms and two original datasets. In Section 3.3, we discuss the conversion of the four axioms, resulting in four axioms that are suitable for obtaining diagnostic datasets from existing corpora. Then, after we introduce the employed original datasets in Section 3.4, we elaborate upon the obtained diagnostic dataset in Section 3.5.

3.1. Motivation

As introduced in Section 2.5, Fang et al. [42, 43] have proposed two strategies for validating whether models fulfilled axioms: a theoretical validation of axiom fulfillment done by hand and an empirical validation with perturbed collections. In Section 3.1.1 we will detail that both of these approaches have become unfeasible in the neural IR era. Subsequently, in Section 3.1.2, we propose a third approach that employs diagnostic datasets, making it suitable for neural models.

3.1.1. Problems in existing methods

The **theoretical approach** to diagnosing non-neural IR models considered a formally expressed axiom and a formally expressed retrieval model [42], as introduced in Section 2.5.1. This approach has proven fruitful for analyzing traditional models such as BM25 and QL in a predictive manner by analyzing their retrieval formulas (without the need to actually conduct empirical experiments) [41–43, 78]. However, looking at the *neural* IR models we study in this work, we can find that this analysis becomes unfeasible for this type of models as they typically encompass more than a hundred up to millions of parameters, as can be seen in Table 3.1.

Model	Amount of parameters
QL	1
BM25	3
RM3	5
DRMM	161
CDSSM	10,877,657
K-NRM	49,763,110
MP-HCNN	~ 71,000,000

Table 3.1: The amount of parameters in traditional and neural IR models: a difference in multiple orders of magnitude. Note that the amount of parameters in neural IR models may vary depending upon the specific architecture. We here show the numbers from [103, 131].

As a second approach, Fang et al. [43] proposed the **empirical collection perturbation approach**. Looking at our research objective, this second strategy can in the neural IR era, no longer "be applied to any retrieval function" as was said in [43]. Collection perturbation operations like word removal may break the syntax and/or semantics of natural language data and hence influence the neural, non-bag-of-words, models [141], which we aim to diagnose. For example, consider how the syntax and/or semantics change if we remove a word from a clause: "why you can not use perturbations". Whereas a bag-of-words model by definition can not pick up that the semantics change or syntax is incorrect, a neural model may be able to. To avoid this potential problem, we will introduce a novel, third method for axiomatic diagnosis.

3.1.2. Axiomatic analysis through diagnostic datasets

Knowing the limitations of existing approaches for axiomatic diagnosis with regard to neural IR, we here propose the use of diagnostic datasets. Different from the analytical and collection perturbation approaches, this approach is 1) feasible for analyzing neural models in an axiomatic manner and 2) does not require relevance-labeled data (of which there is a shortage in neural IR as introduced in Section 1.1).

In short, diagnostic datasets comprise of instances that test whether a model fulfills an axiom or not. We therefore define an axiomatic diagnosis as an empirical evaluation of IR models on a whole dataset of such instances. Using a diagnostic dataset of sufficient size, an axiomatic diagnosis can capture potentially significant differences between IR models in terms of axiom fulfillment. To create diagnostic datasets, we will—different from the collection perturbation approach—typically not engineer artificial dataset instances so that documents fulfill certain conditions posed by an axiom. Instead, we focus on adapting axioms and keep the queries and documents of each instance as is. We can then obtain diagnostic instances by searching for existing queries and pairs or triplets of documents that already fulfill conditions posed by the adapted axioms, as will be discussed next.

The need for axiom conversion

Due to the simple (e.g. one-term queries), though strict (e.g. exact document length equality) definition of axioms, a typical instance in a benchmark dataset (generally consisting of a multi-term query, and a set of typically a multitude of documents) does not meet the conditions imposed by an axiom. In fact, as will be seen in Section 3.3.1, the conditions in the original axioms considered in this work are fulfilled by hardly any instances in the employed datasets. Hence, we can not obtain diagnostic datasets of sufficient size from the considered original datasets using these axioms. To solve this problem, we propose a conversion of strict axioms to a version that poses conditions more likely to be found in existing corpora, as was also found necessary in [53]. We will further discuss this procedure of axiom conversion (and subsequently how to obtain diagnostic datasets) in Section 3.2.

3.2. Creating Diagnostic Datasets

Here we introduce the procedure to create diagnostic datasets. This methodology requires an axiom and an original dataset (i.e. an existing test collection) as input, and then consists of several consecutive steps to come to a diagnostic dataset. In Section 3.2.1, we discuss the conversion of axioms and in Section 3.2.2, we elaborate upon the subsequent creation of a diagnostic dataset.

3.2.1. Converting axiom representations

As introduced in the previous section, each of the established axioms needs to be converted into a form that is suitable for identifying to what extent a dataset instance represents the axiom. We propose two steps:

Step 1 an *extension* of each axiom in order to use realistic query and document sizes;

Step 2 a *relaxation* of extended axioms such that the strictly defined query and document relations are relaxed to enable selection and generation of sufficient amounts of data.

Step 1 allows us to move from one- or two-term queries to arbitrary query lengths and from two- or three-document instances to any number of documents. Formally, we go from e.g. a single-term query $\boldsymbol{q} = \{w\}$ to a multi-term query $\boldsymbol{q} = \{w_1, w_2, ..., w_{|\boldsymbol{q}|}\}$ and from e.g. two documents $\boldsymbol{d_1}, \boldsymbol{d_2}$ to any pair of documents $\boldsymbol{d_i}, \boldsymbol{d_j}$.

Step 2 allows us to make use of query/document pairings that *approximately* fulfill a particular relationship. For example we would go from strict document length equality to a parameterized version, or formally: from $|\boldsymbol{d_i}| = |\boldsymbol{d_j}|$ to $abs(|\boldsymbol{d_i}| - |\boldsymbol{d_j}|) \leq \delta$. Moreover, in our relaxation, we often make use of the fact that we now can have more than one or two query terms. For instance, we can relax conditions such as the count of a term in a document ($c(w, \boldsymbol{d})$) that should hold for each term (e.g. $\forall\, w \in \boldsymbol{q}, c(w, \boldsymbol{d_i}) > c(w, \boldsymbol{d_j})$) to hold for at least one term (e.g. $\sum_{w \in \boldsymbol{q}} c(w, \boldsymbol{d_i}) > \sum_{w \in \boldsymbol{q}} c(w, \boldsymbol{d_j})$). However, such relaxations come at the risk of the axiom no longer representing the idea behind the original axiom. Hence, subsequent modifications (which we here also view as part of the relaxation) are sometimes required to retain the original axiom's intuition in its extended and relaxed version. A graphical overview of the conversion of axiom representations is displayed in Fig. 3.1.

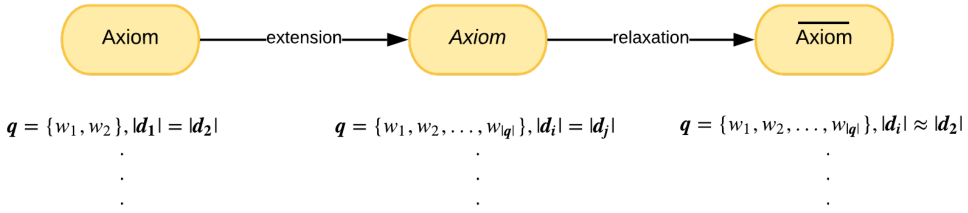

Axiom
extension
Axiom
relaxation
$\overline{\text{Axiom}}$
$\boldsymbol{q} = \{w_1, w_2\}, |\boldsymbol{d_1}| = |\boldsymbol{d_2}|$
$\vdots$
$\boldsymbol{q} = \{w_1, w_2, \ldots, w_{|\boldsymbol{q}|}\}, |\boldsymbol{d_i}| = |\boldsymbol{d_j}|$
$\vdots$
$\boldsymbol{q} = \{w_1, w_2, \ldots, w_{|\boldsymbol{q}|}\}, |\boldsymbol{d_i}| \approx |\boldsymbol{d_2}|$
$\vdots$

Figure 3.1: Graphical representation of axiom *extension* and *relaxation* with part of the `TFC1` conversion shown as an example.

3.2.2. Obtaining diagnostic instances

After the steps for obtaining an extended, relaxed variant of an axiom, we now describe how to obtain a diagnostic dataset for it. Given a corpus with standard pre-processing applied, we determine the number of instances the (i) original axiom, (ii) extended axiom and (iii) relaxed & extended axiom can be found in it. As the axioms are defined over retrieval status value scores (instead of relevance labels), we do not require relevance judgments and, almost any dataset is suitable as source dataset. We can then sample queries and document pairs/triplets at will and keep those in our diagnostic datasets that satisfy our axioms. A graphical overview of the extraction of diagnostic datasets is displayed in Fig. 3.2.

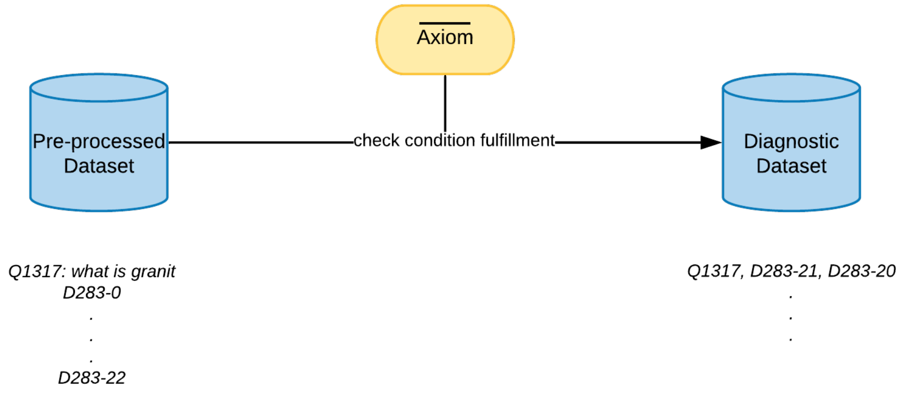


Figure 3.2: Graphical representation of obtaining a diagnostic datasetfrom a pre-processed dataset through checking if instances fulfill conditions raised in an (extended, relaxed) axiom. We show an example for the TFC1 axiom: $D283$-21 should, for query $Q1317$, be ranked higher than $D283$-20.

Next to extracting diagnostic instances from original datasets, we also propose a second approach for obtaining diagnostic instances. This approach enables diagnosis of axioms for which the conditions are fulfilled by too few instances in the original dataset. We thereby turn to adapting documents just like the collection perturbation approach as proposed by Fang et al. [43]. However, we only use this approach as a last resort and use no more operations than strictly necessary to resemble the setting as prescribed by the axiom. We thereby aim to minimize the impact of changes that may influence the deep models, as discussed in Section 3.1.1. Figure 3.3 displays the complete pipeline for creating a diagnostic dataset from an original dataset including both approaches employed in this thesis (diagnostic instance extraction and artificial instance generation).

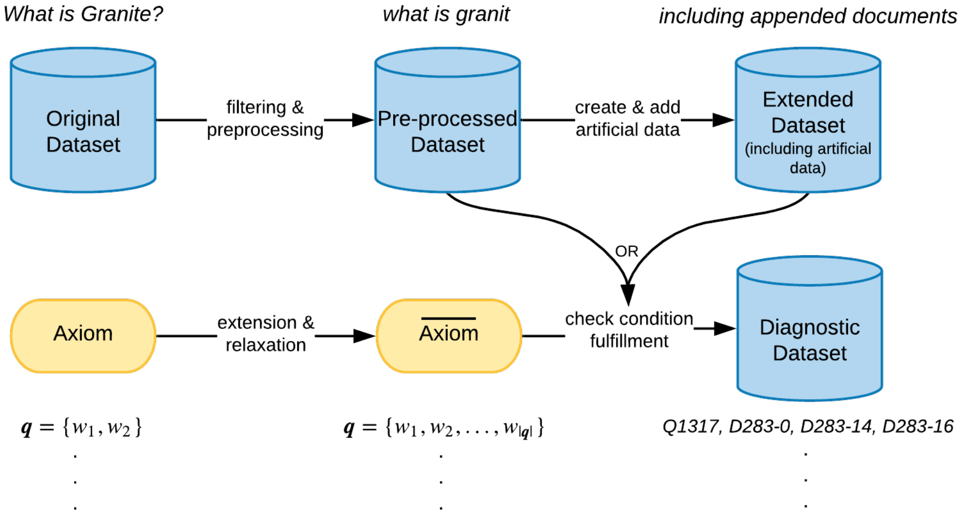

What is Granite?
what is granit
including appended documents
Original Dataset
filtering & preprocessing
Pre-processed Dataset
create & add artificial data
Extended Dataset (including artificial data)
OR
Axiom
extension & relaxation
Axiom
check condition fulfillment
Diagnostic Dataset
$q = \{w_1, w_2\}$
$q = \{w_1, w_2, \ldots, w_{|q|}\}$
Q1317, D283-0, D283-14, D283-16

Figure 3.3: Overview of the diagnostic dataset creation pipeline. In *italics*, we show an example for the $\overline{\text{TFC2}}$ axiom from WikiPassageQA, and refer to appended documents as an example of artifical data (for $\overline{\text{LNC2}}$).

3.3. Axioms Covered in Our Experiments

More than twenty IR axioms[1] have been proposed by now. We have selected four among those, and converted them for our purpose of diagnostic dataset creation. In the following, we will first explain why these axioms were selected in Section 3.3.1. Then, for each axiom, we provide an extension and relaxation in the subsequent sections (Section 3.3.2, 3.3.3, 3.3.4, 3.3.5 resp.).

3.3.1. Axiom selection

We have selected four axioms among those in [42, 43, 114]. Two of the axioms (TFC1 and TDC) were selected as they are expected to capture a fair amount of relevance signals, while intuitively being present in existing datasets. Whereas the intuition behind TFC1 is that models should favour documents with larger query term counts, the intuition behind the M-TDC axiom is that the document frequency of query terms should also be accounted for. Hence, the two axioms combined, essentially represent the TF-IDF statistic. Although the limitations of TF-IDF are well known (e.g. ignoring word orders which may carry syntactic and semantic relationships [72]), the statistic has been a pervasive heuristic in a range of IR models developed over time: TF-IDF is a standard part of pervasive traditional models such as BM25 and language modelling approaches (e.g. QL with Dirichlet Prior smoothing as an IDF-like component [140]) and can provide a strong baseline when combined with n-grams [141]. Some more novel, deep, retrieval approaches have also included explicit TF and/or IDF like characteristics. For example, [22]—which employs siamese convolutional neural networks to learn representations of questions and candidate answer passages—concatenates TF information to QA pairs before feeding them to a feedforward network to produce a relevance score. In [33], IDF information is used in training a neural word embedding. Hence, we expect the two selected axioms provide a good starting point for diagnosing adherence to heuristics that incur retrieval effectiveness.

We have selected a third axiom (TFC2) that does not—like most of the proposed axioms—prescribe that one document should receive a higher score than another, but rather constraints that the difference in scores between a pair of documents should be larger than the difference between another pair of documents. Specifically, the intuition behind the axiom is that the (positive) impact of an increase in TF (on the retrieval status value) should decrease with increasing TF. We include this axiom to show our methodology can handle such axioms as well.

Finally, we selected a fourth axiom (LNC2) to showcase how we can generate a diagnostic dataset from an existing corpus through creating artificial data, rather than using existing data (in which we expect the specific axiom to not (or hardly) be present. Specifically, the intuition behind the axiom is that models should avoid over-penalizing a long document.

An overview of the selected axioms and our motivation to include them is given in Table 3.2.

Axiom	Intuition	Reason for inclusion
TFC1	favour a document with a larger count of a query term	encapsulates part of a pervasive statistic
TFC2	to ensure the impact of an increase in query term count decreases as the count increases	describes a different relationship than most axioms
M-TDC	to assign higher weights to discriminative terms	encapsulates part of a pervasive statistic
LNC2	to avoid over-penalizing a long document	has conditions that are typically not fulfilled by instances in an IR collection

Table 3.2: The intuition behind and the reason for inclusion per axiom covered in this thesis.

[1] An overview of these axioms can be found at https://www.eecis.udel.edu/~hfang/AX.html.

3.3.2. TFC1: extension and relaxation

The TFC1 axiom [42] favours documents with more occurrences of a query term and is formally defined as expressed in Eq. 3.1.

$$\begin{array}{ll} \text{Assume} & \boldsymbol{q} = \{w\} \text{ and } |\boldsymbol{d_1}| = |\boldsymbol{d_2}|, \\ \text{If} & c(w, \boldsymbol{d_1}) > c(w, \boldsymbol{d_2}), \\ \text{Then} & S(\boldsymbol{d_1}, \boldsymbol{q}) > S(\boldsymbol{d_2}, \boldsymbol{q}) \end{array} \tag{3.1}$$

We now *extend* this axiom to multi-term queries and multiple documents, resulting in Eq. 3.2.

$$\begin{array}{ll} \text{Assume} & \boldsymbol{q} = \{w_1, w_2, ..., w_{|\boldsymbol{q}|}\} \text{ and } |\boldsymbol{d_i}| = |\boldsymbol{d_j}|, \\ \text{If} & c(w, \boldsymbol{d_i}) > c(w, \boldsymbol{d_j}) \text{ for all } w \in \boldsymbol{q}, \\ \text{Then} & S(\boldsymbol{d_i}, \boldsymbol{q}) > S(\boldsymbol{d_j}, \boldsymbol{q}) \end{array} \tag{3.2}$$

Subsequently, we *relax* it to incorporate documents of approximately the same length, as defined with a parameter $\delta_{\texttt{TFC1}}$. This parameter should be set depending on the original document corpus and retrieval task. Additionally, we *relax* the constraint that $\boldsymbol{d_i}$ must have a larger count than $\boldsymbol{d_j}$ for *every* query term. Instead, we now require that there is at least one query term with a higher term count in $\boldsymbol{d_i}$ and that there is *no* query term for which $\boldsymbol{d_j}$ has a higher count than $\boldsymbol{d_i}$. Incorporating these relaxations brings us to Eq. 3.3, or as will refer to it now: $\overline{\texttt{TFC1}}$.

$$\begin{array}{ll} \text{Assume} & \boldsymbol{q} = \{w_1, w_2, ..., w_{|\boldsymbol{q}|}\} \text{ and } abs(|\boldsymbol{d_i}| - |\boldsymbol{d_j}|) \leq \delta_{\texttt{TFC1}}, \\ \text{If} & c(w, \boldsymbol{d_i}) \geq c(w, \boldsymbol{d_j}) \text{ for all } w \in \boldsymbol{q} \text{ and } \sum_{w \in \boldsymbol{q}} c(w, \boldsymbol{d_i}) > \sum_{w \in \boldsymbol{q}} c(w, \boldsymbol{d_j}), \\ \text{Then} & S(\boldsymbol{d_i}, \boldsymbol{q}) > S(\boldsymbol{d_j}, \boldsymbol{q}) \end{array} \tag{3.3}$$

3.3.3. TFC2: extension and relaxation

Axiom TFC2 [42] encapsulates the intuition that an increase in retrieval status value due to an increase in term count becomes smaller as the absolute term count increases. Eq. 3.4 contains a formal expression of the axiom in which the absolute term count of w in $\boldsymbol{d_1}$ is smallest and in $\boldsymbol{d_3}$ is largest.

$$\begin{array}{ll} \text{Assume} & \boldsymbol{q} = \{w\} \text{ and } |\boldsymbol{d_1}| = |\boldsymbol{d_2}| = |\boldsymbol{d_3}|, \\ \text{If} & c(w, \boldsymbol{d_1}) > 0,\ c(w, \boldsymbol{d_2}) - c(w, \boldsymbol{d_1}) = 1 \text{ and } c(w, \boldsymbol{d_3}) - c(w, \boldsymbol{d_2}) = 1, \\ \text{Then} & S(\boldsymbol{d_2}, \boldsymbol{q}) - S(\boldsymbol{d_1}, \boldsymbol{q}) > S(\boldsymbol{d_3}, \boldsymbol{q}) - S(\boldsymbol{d_2}, \boldsymbol{q}) \end{array} \tag{3.4}$$

Again, we first define an extended variant of TFC2 that considers multi-term queries and, in this case, any triplet of documents. Formally this results in:

$$\begin{array}{ll} \text{Assume} & \boldsymbol{q} = \{w_1, w_2, ..., w_{|\boldsymbol{q}|}\} \text{ and } |\boldsymbol{d_i}| = |\boldsymbol{d_j}| = |\boldsymbol{d_k}|, \\ \text{If} & c(w, \boldsymbol{d_i}) > 0,\ c(w, \boldsymbol{d_j}) - c(w, \boldsymbol{d_i}) = 1 \text{ and } c(w, \boldsymbol{d_k}) - c(w, \boldsymbol{d_j}) = 1 \text{ for all } w \in \boldsymbol{q}, \\ \text{Then} & S(\boldsymbol{d_j}, \boldsymbol{q}) - S(\boldsymbol{d_i}, \boldsymbol{q}) > S(\boldsymbol{d_k}, \boldsymbol{q}) - S(\boldsymbol{d_j}, \boldsymbol{q}) \end{array} \tag{3.5}$$

For our relaxed version of the axiom, we consider $\boldsymbol{q} = \{w_1, w_2, \ldots, w_{|\boldsymbol{q}|}\}$ and $|\boldsymbol{d_i}| \approx |\boldsymbol{d_j}| \approx |\boldsymbol{d_k}|$, i.e. $\max_{\boldsymbol{d_1}, \boldsymbol{d_2} \in \{\boldsymbol{d_i}, \boldsymbol{d_j}, \boldsymbol{d_k}\}}(|\boldsymbol{d_1}| - |\boldsymbol{d_2}|) \leq |\delta_{\texttt{TFC2}}|$. Furthermore, we now constrain that every document has to contain at least one query term (instead of all query terms) and no longer restrict the differences in term count to be exactly 1. This leads to the constraints $\sum_{w \in \boldsymbol{q}} c(w, \boldsymbol{d_k}) > \sum_{w \in \boldsymbol{q}} c(w, \boldsymbol{d_j}) > \sum_{w \in \boldsymbol{q}} c(w, \boldsymbol{d_i}) > 0$ and $c(w, \boldsymbol{d_j}) - c(w, \boldsymbol{d_i}) = c(w, \boldsymbol{d_k}) - c(w, \boldsymbol{d_j})$ for all $w \in \boldsymbol{q}$. The latter constraint does not mean that the difference has to be the same for every query term, instead we enforce this equality in term count difference on a term level. If these constraints hold, then according to $\overline{\texttt{TFC2}}$, $S(\boldsymbol{d_j}, \boldsymbol{q}) - S(\boldsymbol{d_i}, \boldsymbol{q}) > S(\boldsymbol{d_k}, \boldsymbol{q}) - S(\boldsymbol{d_j}, \boldsymbol{q})$.

$$\begin{array}{ll} \text{Assume} & \boldsymbol{q} = \{w_1, w_2, ..., w_{|\boldsymbol{q}|}\} \text{ and } \max_{\boldsymbol{d_1},\boldsymbol{d_2} \in \{\boldsymbol{d_i},\boldsymbol{d_j},\boldsymbol{d_k}\}}(|\boldsymbol{d_1}| - |\boldsymbol{d_2}|) \leq |\delta_{\texttt{TFC2}}|, \\ \text{If} & \sum_{w \in \boldsymbol{q}} c(w,\boldsymbol{d_k}) > \sum_{w \in \boldsymbol{q}} c(w,\boldsymbol{d_j}) > \sum_{w \in \boldsymbol{q}} c(w,\boldsymbol{d_i}) > 0 \\ & \text{and } c(w,\boldsymbol{d_j}) - c(w,\boldsymbol{d_i}) = c(w,\boldsymbol{d_k}) - c(w,\boldsymbol{d_j}) \text{ for all } w \in \boldsymbol{q}, \\ \text{Then} & S(\boldsymbol{d_j},\boldsymbol{q}) - S(\boldsymbol{d_i},\boldsymbol{q}) > S(\boldsymbol{d_k},\boldsymbol{q}) - S(\boldsymbol{d_j},\boldsymbol{q}) \end{array} \quad (3.6)$$

3.3.4. M-TDC: extension and relaxation

The TDC axiom was originally proposed by Fang et al. [42] to favour documents with more occurrences of less popular query terms in the collection. Shi et al. modified the TDC axiom to M-TDC [114] to fix[2] some undesired behaviour. Formally, M-TDC is defined as displayed in Eq. 3.7. Note that w_1 is rarer in the corpus than w_2.

$$\begin{array}{ll} \text{Assume} & \boldsymbol{q} = \{w_1, w_2\}, |\boldsymbol{d_1}| = |\boldsymbol{d_2}|, c(w_1,\boldsymbol{d_1}) = c(w_2,\boldsymbol{d_2}) \text{ and } c(w_2,\boldsymbol{d_1}) = c(w_1,\boldsymbol{d_2}), \\ \text{If} & idf(w_1) \geq idf(w_2) \text{ and } c(w_1,\boldsymbol{d_1}) \geq c(w_1,\boldsymbol{d_2}), \\ \text{Then} & S(\boldsymbol{d_1},\boldsymbol{q}) \geq S(\boldsymbol{d_2},\boldsymbol{q}) \end{array} \quad (3.7)$$

We then define $\overline{\texttt{M-TDC}}$ for multi-term queries and any pair of documents, resulting in Eq. 3.8.

$$\begin{array}{ll} \text{Assume} & \boldsymbol{q} = \{w_1, w_2, ..., w_{|q|}\}, |\boldsymbol{d_i}| = |\boldsymbol{d_j}|, \ c(w_a,\boldsymbol{d_i}) = c(w_b,\boldsymbol{d_j}) \text{ and} \\ & c(w_b,\boldsymbol{d_i}) = c(w_a,\boldsymbol{d_j}) \text{ for all } w_a, w_b \in \boldsymbol{q} \text{ with } w_a \neq w_b, \\ \text{If} & idf(w_a) \geq idf(w_b) \text{ and } c(w_a,\boldsymbol{d_i}) \geq c(w_a,\boldsymbol{d_j}) \text{ for all } w_a, w_b \in \boldsymbol{q} \text{ with} \\ & w_a \neq w_b, \\ \text{Then} & S(\boldsymbol{d_i},\boldsymbol{q}) \geq S(\boldsymbol{d_j},\boldsymbol{q}) \end{array} \quad (3.8)$$

Now we turn to the relaxation of our extended version of M-TDC. Note that all conditions in Eq. 3.8 hold if (and only if) $c(w,\boldsymbol{d_i}) = c(w,\boldsymbol{d_j})$ for all $w \in \boldsymbol{q}$, due to the "Assume" conditions. Evidently instances that fulfill this condition do not allow to diagnose if models do favour documents with more occurrences of less popular query terms as the intuition behind M-TDC encompasses. Hence, we relax this strict condition and only constrain that the total sum of query terms should be equal, but also constrain $\boldsymbol{d_i}$ and $\boldsymbol{d_j}$ to differ in at least one query term count, resulting in the "Assume" conditions in the first two lines of Eq. 3.9.

Now we have to make sure that $\boldsymbol{d_i}$, compared to $\boldsymbol{d_j}$, will always have a larger (or equal) count of query terms that are less popular in the collection - just like the original axiom regarding two terms prescribed that $\boldsymbol{d_1}$, compared to $\boldsymbol{d_2}$, had a larger count of the less popular query term. So, if $idf(w_a) \geq idf(w_b)$ (subcondition 1 in Eq. 3.9) then $c(w_a,\boldsymbol{d_i}) > c(w_a,\boldsymbol{d_j})$ should hold as well (subcondition 2). However, since we are now considering queries that can consist of multiple terms, we have to avoid an issue. If a word w_a that is less popular in the collection than a word w_b, but a query $\boldsymbol{q}$ contains the word w_b more often than w_a, we may not be sure whether it is more important to have a higher count of w_a or w_b - should we focus on the idf value or on the frequency of the term in the query? Hence, we also constrain that $c(w_a,\boldsymbol{q}) \geq c(w_b,\boldsymbol{q})$ to avoid running into this issue (subcondition 3).

Now, we have to avoid a second issue. For example, consider a three term query $\boldsymbol{q} = \{w_1, w_2, w_3\}$ with $idf(w_1) < idf(w_2) < idf(w_3)$, if $c(w_1,\boldsymbol{d_i}) = 10$, $c(w_2,\boldsymbol{d_i}) = c(w_3,\boldsymbol{d_i}) = 0$ and $c(w_1,\boldsymbol{d_j}) = 0$, $c(w_2,\boldsymbol{d_j}) = c(w_3,\boldsymbol{d_j}) = 5$, we can obtain that all conditions we had so far are fulfilled. However, this contradicts the TFC3 axiom that states that a document that contains more distinct query terms (in our case $\boldsymbol{d_j}$) should be ranked above a document containing less distinct query terms (in our case $\boldsymbol{d_i}$) [42, 43]. Hence, we need to constrain that the count of both terms w_a and w_b are swapped across documents $\boldsymbol{d_i}$ and $\boldsymbol{d_j}$ (subcondition 4). The latter is in essence the fix of M-TDC over TDC as presented in [114].

Finally, as a relaxation, we only enforce these conditions to hold for query terms for which $\boldsymbol{d_i}$ and $\boldsymbol{d_j}$ have a different count and ignore terms for which they have an equal count. Note that we know that $\boldsymbol{d_i}$ and $\boldsymbol{d_j}$ differ in at least one query term count (as enforced in the second line of Eq. 3.9). Ultimately, the formal definition of $\overline{\texttt{M-TDC}}$ becomes:

[2] Specifically, the fix was to avoid the issue that the original TDC axiom contradicts TFC1 and TFC3 (the latter is wrongly listed as TFC2 in [114]).

$$\begin{array}{ll}
\text{Assume} & \boldsymbol{q} = \{w_1, w_2, ..., w_{|\boldsymbol{q}|}\},\ abs(|\boldsymbol{d_i}| - |\boldsymbol{d_j}|) \leq \delta_{\texttt{M-TDC}}, \\
 & \sum_{w\in\boldsymbol{q}} c(w,\boldsymbol{d_i}) = \sum_{w\in\boldsymbol{q}} c(w,\boldsymbol{d_j}) \text{ and } \exists w \in \boldsymbol{q} \text{ so that } c(w,\boldsymbol{d_i}) \neq c(w,\boldsymbol{d_j}), \\
\text{If} & \text{for all } w \in \boldsymbol{q} \text{ for which } c(w,\boldsymbol{d_i}) \neq c(w,\boldsymbol{d_j}),\ w = w_a \text{ or } w = w_b \\
 & \text{with } w_a, w_b \in \boldsymbol{q} \text{ and } w_a \neq w_b \text{ such that:} \\
 & \quad 1)\ idf(w_a) \geq idf(w_b), \\
 & \quad 2)\ c(w_a,\boldsymbol{d_i}) > c(w_a,\boldsymbol{d_j}), \\
 & \quad 3)\ c(w_a,\boldsymbol{q}) \geq c(w_b,\boldsymbol{q}), \\
 & \quad 4)\ c(w_a,\boldsymbol{d_i}) = c(w_b,\boldsymbol{d_j}) \text{ and } c(w_b,\boldsymbol{d_i}) = c(w_a,\boldsymbol{d_j}), \\
\text{Then} & S(\boldsymbol{d_i},\boldsymbol{q}) \geq S(\boldsymbol{d_j},\boldsymbol{q})
\end{array} \tag{3.9}$$

3.3.5. LNC2: extension and relaxation

The LNC2 [43] axiom prescribes that over-penalizing long documents should be avoided: if a document is replicated k times, its retrieval status score should not be lower than that of its un-replicated variant. It should be noted that the axiom was defined under the assumption that redundancy is not an issue, which we also follow here. Formally the axiom is defined as follows[3]:

$$\begin{array}{ll}
\text{Assume} & \boldsymbol{q} = \{w\},\ k \in \mathbb{N},\ k > 1,\ |\boldsymbol{d_1}| = k \times |\boldsymbol{d_2}| \text{ and } c(w,\boldsymbol{d_1}) > 0, \\
\text{If} & c(w,\boldsymbol{d_1}) = k \times c(w,\boldsymbol{d_2}) \text{ for all } w \in \boldsymbol{d_1}, \\
\text{Then} & S(\boldsymbol{d_1},\boldsymbol{q}) \geq S(\boldsymbol{d_2},\boldsymbol{q})
\end{array} \tag{3.10}$$

We then define $\overline{\texttt{LNC2}}$ by defining $\boldsymbol{q}$ for multi-term queries and documents $\boldsymbol{d_i}$ and $\boldsymbol{d_j}$, resulting in Eq. 3.11.

$$\begin{array}{ll}
\text{Assume} & \boldsymbol{q} = \{w_1, w_2, ..., w_{|\boldsymbol{q}|}\},\ k \in \mathbb{N},\ k > 1,\ |\boldsymbol{d_i}| = k \times |\boldsymbol{d_j}| \text{ and} \\
 & c(w,\boldsymbol{d_i}) > 0 \text{ for all } w \in \boldsymbol{q}, \\
\text{If} & c(w,\boldsymbol{d_i}) = k \times c(w,\boldsymbol{d_j}) \text{ for all } w \in \boldsymbol{d_j}, \\
\text{Then} & S(\boldsymbol{d_i},\boldsymbol{q}) \geq S(\boldsymbol{d_j},\boldsymbol{q})
\end{array} \tag{3.11}$$

Now, we only relax the condition that document $\boldsymbol{d_i}$ should contain every query term: we only enforce it to contain one query term. Note that we do not relax $\boldsymbol{d_j}$ to contain any words that are not present in $\boldsymbol{d_i}$, to avoid the potential impact if such words are semantically similar to query terms[4]. Hence, the formal definition of $\overline{\texttt{LNC2}}$ becomes:

$$\begin{array}{ll}
\text{Assume} & \boldsymbol{q} = \{w_1, w_2, ..., w_{|\boldsymbol{q}|}\},\ k \in \mathbb{N},\ k > 1,\ |\boldsymbol{d_i}| = k \times |\boldsymbol{d_j}| \text{ and } \sum_{w\in\boldsymbol{q}} c(w,\boldsymbol{d_i}) > 0, \\
\text{If} & c(w,\boldsymbol{d_i}) = k \times c(w,\boldsymbol{d_j}) \text{ for all } w \in \boldsymbol{d_j}, \\
\text{Then} & S(\boldsymbol{d_i},\boldsymbol{q}) \geq S(\boldsymbol{d_j},\boldsymbol{q})
\end{array} \tag{3.12}$$

[3] Note that the original LNC2 axiom as defined in [42, 43] does not explicitly enforce $c(w,\boldsymbol{d_1}) > 0$ in the formula. However, from the analyses conducted in the paper, we conclude that this requirements was meant to be part of the axiom. For example, it was found that BM25 fulfills the LNC2 axiom conditionally, whereas if $\boldsymbol{q} = \{w\}$ and $c(w,\boldsymbol{d_1}) = 0$, i.e. $\boldsymbol{d}_1$ and therefore $\boldsymbol{d_2}$ does not contain any query terms, holds, BM25 would always fulfill the axiom (since $S(\boldsymbol{d_1},\boldsymbol{q}) = S(\boldsymbol{d_2},\boldsymbol{q}) = 0$). A similar argument can be made to add the $c(w,\boldsymbol{d_1}) > 0$ constraint to the M-TDC axiom, however for simplicity we have not included this constraint in the initial formula, as it is enforced in the relaxation (through the "Assume" conditions in Eq. 3.9).

[4] For completeness, we note that this would then contradict the semantic term matching constraints (the STMC axioms) proposed in [41].

3.4. Employed Datasets

As previously mentioned, the field of neural IR faces a lack of large scale public datasets [29] required for training neural IR models. Recently, some datasets have been released that address this issue. In this work we have employed two of such large scale datasets: `WikiPassageQA` (released in early Autumn 2018) and `MSMarco` (released in October 2018). We will subsequently discuss each of them in the following sections 3.4.1 and 3.4.2 and finally compare them in Section 3.4.3.

3.4.1. `WikiPassageQA`

The `WikiPassageQA` corpus has been developed for the *answer passage retrieval* task: given a query (a question) and a Wikipedia document (more concretely, all passages making up that document), rank the passages such that those containing the answer to the question are ranked on top. It can thus be seen as a variant of the QA task. `WikiPassageQA` contains over 4,000 queries on the top 800 Wikipedia documents from the Open Wikipedia Ranking[5], making it the "only large data set with long passages as answers for thousands of non-factoid questions in the open domain" at release time as stated by Cohen et al. [22]. The dataset consists of labeled non-factoid answer passage retrieval tasks, of which two examples are provided in Table 3.3. Cohen et al. [22] also provide the results obtained with two baselines, three traditional IR models and five neural IR models on the corpus. In the following, we elaborate upon how we filtered and pre-processed the dataset, discuss how the dataset was created as well as its characteristics and finally discuss adopted evaluation metrics.

Query 4114:
Question: Why is Japan so densely populated?
Document ID: 496
Document Name: Japan.html
Answer Passage(s):

Passage 17 The main islands, from north to south, are Hokkaido, Honshu, Shikoku and Kyushu. The Ryukyu Islands, which include Okinawa, are a chain to the south of Kyushu. Together they are often known as the Japanese archipelago. About 73% of Japan is forested, mountainous, and unsuitable for agricultural, industrial, or residential use. As a result, the habitable zones, mainly located in coastal areas, have extremely high population densities. Japan is one of the most densely populated countries in the world.

Passage 41 Japan is the second-largest agricultural product importer in the world. Rice, the most protected crop, is subject to tariffs of 777.7%. In 1996, Japan ranked fourth in the world in tonnage of fish caught. Japan captured 4,074,580 metric tons of fish in 2005, down from 4,987,703 tons in 2000, 9,558,615 tons in 1990, 9,864,422 tons in 1980, 8,520,397 tons in 1970, 5,583,796 tons in 1960 and 2,881,855 tons in 1950. In 2003, the total aquaculture production was predicted at 1,301,437 tonnes. In 2010, Japan's total fisheries production was 4,762,469 fish.

Query 2402:
Question: What is the structure of Australia's members of parliament?
Document ID: 400
Document Name: Member_of_parliament.html
Answer Passage(s):

Passage 0 A Member of Parliament is the representative of the voters to a parliament. In many countries with bicameral parliaments, this category includes specifically members of the lower house, as upper houses often have a different title. Members of parliament tend to form parliamentary groups with members of the same political party. The Westminster system is a democratic parliamentary system of government modelled after the politics of the United Kingdom. This term comes from the Palace of Westminster, the seat of the Parliament of the United Kingdom. A member of parliament is a member of the House of Representatives, the lower house of the Commonwealth parliament. Members may use "MP" after their names; "MHR" is not used, although it was used as a post-nominal in the past.

Passage 1 A member of the upper house of the Commonwealth parliament, the Senate, is known as a "Senator". In the Australian states of New South Wales, Victoria and South Australia, a Member of the Legislative Assembly or "lower house", may also use the post-nominal "MP." Members of the Legislative Council use the post-nominal "MLC." Members of the Jatiyo Sangshad, or National Assembly, are elected every five years and are referred to in English as members of Parliament. The assembly has directly elected 300 seats, and further 50 reserved selected seats for women. The Parliament of Canada consists of the monarch, the Senate, and the House of Commons.

Table 3.3: Sample questions and a subset of the associated passages in the non pre-processed `WikiPassageQA` dataset, annotated answer passages are marked green , non-answer passages are marked red . Models are given a question and the set of all passages that make up the Wikipedia page related to the question and have to rank answer passages above non-answer passages in their result list. Figure adapted from [22].

[5] See http://wikirank.di.unimi.it/.

Filtering & Pre-processing

WikiPassageQA has been released with a pre-defined train/dev/test split that we maintain in our work [22]. In terms of pre-processing, we apply stemming[6] but not stopword removal, as the latter may actually remove informative terms from the text such as the question words *what* and *why*. To be able to use the dataset in the Indri retrieval toolkit, we only maintain alphanumeric characters (and single spaces). Furthermore, we have identified various special cases in the WikiPassageQA dataset and list below how we handled each of them:

- We note that there are no questions related to the document with docid 188, but keep it in the corpus, as was done in the benchmark experiments;
- We note that there is an extremely large document in the corpus (the passage with pid 18 in the document with docid 403 contains 1332 words)[7], but keep it in the corpus, as was done in the benchmark experiments;
- The questions with qid 4149 (dev), 4148 and 1315 (both in train) only contain the symbols "{}" and no actual question and hence we remove these questions from our dataset;
- The question "How does the WTO function?" is part of both the training (qid 3731) and test set (qid 3732) but has different answers in both cases. We have removed this ambiguous instance from both the train and test dataset splits;
- We do not remove any special instances that only have passages that are answers (qid 903, 3993, 3727, 3728, 3729).

From here on we refer to WikiPassageQA as being the filtered, pre-processed version of the dataset that was obtained after applying the enlisted filtering and pre-processing steps (unless explicitly stated otherwise).

Creation & Characteristics

WikiPassageQA consists of 861 Wikipedia documents, split into passages of six sentences. For splitting the documents into passages, a strided window was employed (returning subsequent passages of six sentences) . Hence, the last passage of the splitted Wikipedia document may contain less than six sentences. This process yield 50,477 unique passages in total—each containing 135.2 words on average (minimum 11, maximum 1332). The 4,186 questions in the dataset were created by crowd-workers employed through Amazon Mechanical Turk[8]. The created questions contain 9.5 terms on average (minimum 2, maximum 39)[9].

The binary passage-level relevance judgments were also sourced from the same crowd-workers that posed the questions and were later validated by a subsequent mechanical turk verification poll. These relevance judgments encompass each passage (of a Wikipedia document) per query (on that Wikipedia document). On average, there are 1.7 relevant passages per question.

Evaluation Metrics

As in [22], we employ mean average precision (MAP), mean reciprocal rank (MRR) and precision at k documents (P@k) to report retrieval effectiveness. In terms of *axiomatic performance*, we report the fraction of diagnostic instances each model satisfies per axiom.

[6] We employed the nltk.stem.SnowballStemmer for the English language.

[7] This passage originates from the Wikipedia page on “The Allies of World War I”, containing a large list (without periods that mark the end of sentences), as, at the time of writing, can still be found at https://en.wikipedia.org/wiki/Allies_of_World_War_I#Leaders.

[8] See https://www.mturk.com/.

[9] Before pre-processing, the shortest query reads: “define Hydroelectricity”, the longest query reads: “Once elected on his right in 1904 to a full term, how could it be argued that the major splits inside President Roosevelt’s republican party probably led to the Democrats return to power 1912 following W. H. Taft Presidency?”.

3.4.2. MS MARCO

MSMarco (MicroSoft MAchine Reading COmprehension) is a large scale dataset focused on machine reading comprehension, question answering, and passage ranking [92]. In this work, we employ the dataset that was prepared for the *passage (re)ranking task*: given a query (a question) and a set of passages, rank the passages such that those that are labeled as relevant are ranked on top. It can hence be seen as a variant of the ad-hoc retrieval task. It contains over 5,000,000 queries users typed into the Bing search engine[10] and the passages are snippets that were extracted from real web documents through Bing[11]. It was created to facilitate the "benchmarking of ML based retrieval models" [92]. A variant of the benchmark will be part of TREC 2019 as an ad-hoc task called the "Deep Learning Track"[12]. Two example queries and two associated passages from the dataset are displayed in Table 3.4.

Query 538699 : Question: wadesboro population Candidate Passage(s):
Passage 4882673 Wadesboro, NC Population and Races. As of 2010-2014, the total population of Wadesboro is 5,711, which is 60.78% more than it was in 2000. The population growth rate is much higher than the state average rate of 21.13% and is much higher than the national average rate of 11.61%.
Passage 1709414 Population of the 100 Largest Urban Places: 1840 (6k) 8. Population of the 100 Largest Urban Places: 1850 (7k) 9. Population of the 100 Largest Urban Places: 1860 (6k) 10. Population of the 100 Largest Urban Places: 1870 (6k) 11. Population of the 100 Largest Urban Places: 1880 (6k) 12. Population of the 100 Largest Urban Places: 1890 (6k) 13. Population of the 100 Largest Urban Places: 1900 (6k) 14. Population of the 100 Largest Urban Places: 1910 (7k) 15. Population of the 100 Largest Urban Places: 1920 (7k) 16. Population of the 100 Largest Urban Places: 1930 (7k
Query 215307: Question: how did the missouri compromise affect massachusetts Candidate Passage(s):
Passage 5241523 The Missouri Compromise of 1820 admitted Missouri as a slave state, and Maine as a free state, to keep the balance of slave/non-slave states equal in Congress. It also established the 36-30 line, and said that slaveery would not be allowed above that line except for in Missouri. So the Compromise set a bunch of rules about slavery, but the big thing it did was it separated Maine from Massachusetts and allowed it to become a free state. Missouri was allowed to become a slave state.
Passage 5241524 The Missouri Compromise of 1820 admitted Missouri as a slave state, and Maine as a free state, to keep the balance of slave/non-slave states equal in Congress. It also established the 36-30 line, and said that slaveery would not be allowed above that line except for in Missouri.he Missouri Compromise created the 36th parallel in the United States, the Mason Dixon Line. The Mason Dixon line was an imaginary line that divided the North and South.

Table 3.4: Sample queries and passages in the non pre-processed MSMarco dataset, annotated answer passages are marked green , non-answer passages are marked red . overview inspired by [22]. Retrieval models are given a question and set of passages and have to rank relevant passages above non-relevant passages in their result list. Figure inspired by [22].

Filtering & Pre-processing

For each question, Nguyen et al. [92] have by now made a set of 1000 associated passages available (a shuffled version of the top 1000 passages obtained through running a standard BM25 on the complete MSMarco document collection). Since such data was not properly available before the start of our experiments (only a top 1000 for the dev set was made available), and the authors encourage the use your own top-k retrieval model to obtain a set of passages associated to a question[13], we have not used these published sets of passages. Instead, we have run a standard QL as was done in [59, 60][14], and obtain a top 50 of passages per question. As will be explained in Section 4.1.2, we obtain 50 rather than 1000 passages per question to avoid experimental issues. We furthermore do not maintain the original train/dev/test split, since we do not have answer labels for the test split. Although these labels are not needed for obtaining diagnostic datasets, we here need them to be able to obtain retrieval effectiveness of models, which we will compare with their axiomatic scores. We extract our own test set from the training set (and remove it from the training set), hence we only employ queries in the train and dev set of the original dataset in this thesis.

For this corpus we have also identified and handled various issues (most of them resulting from our top-k retrieval approach):

[10] See https://www.bing.com.

[11] No further details have been made available on how this was specifically done.

[12] See https://trec.nist.gov/pubs/call2019.html.

[13] See https://github.com/dfcf93/MSMARCO/issues/21.

[14] Although BM25 has been more widely adopted in top-k retrieval, we here follow [59, 60] and employ QL, which is also the default model in the employed Indri toolkit and fulfills more of the original axioms than BM25.

- More than one-third of the questions (38.61 %) did not have any passage that was annotated as relevant in the document collection and hence, these questions have been removed;
- In case QL did not return any of the annotated answer passages for a question in its top 50 of passages for that question, we added this passage to the 50 retrieved passages;
- Our QL ranking did not return a ranking for 3 questions, for which no query term was present in any document in the corpus.[15] We have removed these three questions from our dataset;
- For another 25 questions, our QL pre-ranker returned less than 50 documents (min. 7, max. 47, average 21.6), all of these concerned two-term queries of which more than half concerned first- and surnames of persons. We have kept such questions in our dataset (as was also done by [101] for different datasets);
- 333 out of the nearly 9 million documents are duplicated versions of other documents in the collection (e.g. the document with docid 1017066 is equal to the document with docid 496964 appended to itself and is hence twice as long as the document with docid 496964), we keep such documents in our corpus.

Creation & Characteristics

The MSMarco dataset provides us with 558,517 queries that were posed by Bing users—each containing 5.99 words on average (minimum 1, maximum 40)[16]. Passages in the set of 8.8 million contain 63.05 words on average (minimum 2, maximum 289)[17].

All relevance judgments come from human judges that, for each query, have only annotated the top 10 passages as retrieved by the Bing stack. Hence, other documents among the 8.8 million passages can also be relevant, i.e. the dataset may very well contain false negatives, albeit is said to have no false *positives*. On average we have 1.06 answers per question.

Evaluation Metrics

The creators of the MSMarco dataset have held out the test set to evaluate models for which they maintain a leaderboard[18]. Since this leaderboard for the passage re-ranking task is based upon the the mean reciprocal rank (MRR) metric, we also use it in our work to report retrieval effectiveness. To obtain more knowledge on differences in models' retrieval effectiveness we also employ MAP and P@k. As stated before, in terms of *axiomatic performance*, we report the fraction of diagnostic instances each model satisfies per axiom.

3.4.3. Comparison

A side-by-side overview of the characteristics of both the WikiPassageQA and the MSMarco dataset is displayed in Table 3.5. Following [22], the displayed statistics are defined on a per-question basis, e.g. we have taken the sum of all candidate passages per question to obtain the total amount of candidate passages.

As can be seen in the table, the characteristics of the WikiPassageQA dataset are similar to the characteristics of the MSMarco (albeit, about two orders of magnitude smaller in the absolute values). However, strong differences arise in the length of questions, answers and documents. The average length of a question or an answer or non-answer passage is nearly twice as long and the average length of an answer passage is nearly thrice as long in WikiPassageQA compared to MSMarco. Additionally, we can find for both datasets that the average length of a (answer or non-answer) passage differs from the average length of an answer passage (See Fig. 3.4). Considering WikiPassageQA, we note that the difference in the average length of all passages (135.46) compared to the average length of answer passages (146.87) can relate to the fact that the answer passages are hardly ever the last passage in a Wikipedia document (See Fig. 3.5) which are the only passages that can contain less than six sentences due to how the dataset was created as discussed in Section 3.4.2.

[15] Specifically, (before and) after pre-processing, they read "tootlesdefinit(ion)" (qid 522517), "hopefullymeaning" (qid 205266) and "standingdefinit(ion)" (qid 502557).

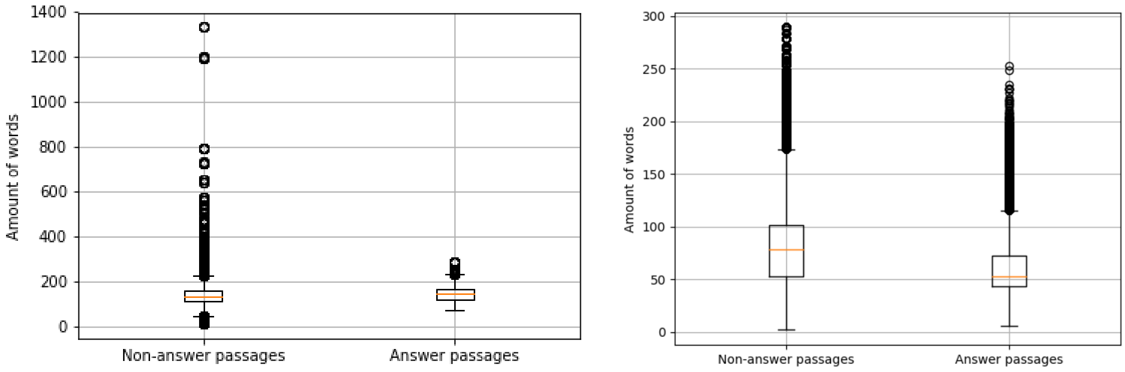
[16] The one-term queries were discussed in the previous paragraph, the queries that contain 40 terms are "the average concentration of a chemical in the air to which a worker can be exposed over a particular period of time (usually eight hours) if referred to as a" (qid 514265) and "'which canadian province has a strong french identity and takes a leading role in developing a new global french technical language? a. ontario b. new brunswick c. manitoba d. quebec" (qid 1006451).

[17] The two shortest documents contain many special characters from languages with a different alphabet (respectively Bulgarian and Malayalam) that have been removed in our pre-processing (pid 5465355, 8550000), the longest document contains the lyrics and a URL to the video of the song "What do you mean" by Justin Bieber (pid 1814137).

[18] See http://www.msmarco.org/leaders.aspx.

Collection	WikiPassageQA	MSMarco
Questions	4,154	558,514
CandidateP	243,489	28,209,143
PosCandidateP	6,947	592,031
NegCandidateP	236,542	27,617,112
PositiveP/CandidateP	0.03	0.02
CandidateP/Query	58.62	50.51
PosCandidateP/Query	1.67	1.06
AvgLenOfQuestion	9.52	5.99
AvgLenOfAnswerP	146.87	59.47
AvgLenOfP	135.46	79.17

Table 3.5: Collection statistics for our pre-processed versions of the `WikiPassageQA` and `MSMarco` datasets. “P” in the first column denotes “Passages”. Statistics refer to numbers on a per-question basis (e.g. the CandidateP contains duplicates as we sum all candidate passages per question which may be about the same Wikipedia document), overview inspired by [22].



(a) Distribution of document length in non-answer passages (left) and answer passages (right) in the `WikiPassageQA` dataset.

(b) Distribution of document length in non-answer passages (left) and answer passages (right) in the `MSMarco` dataset.

Figure 3.4: Distribution of document length in non-answer passages and answer passages in `WikiPassageQA` and `MSMarco`.

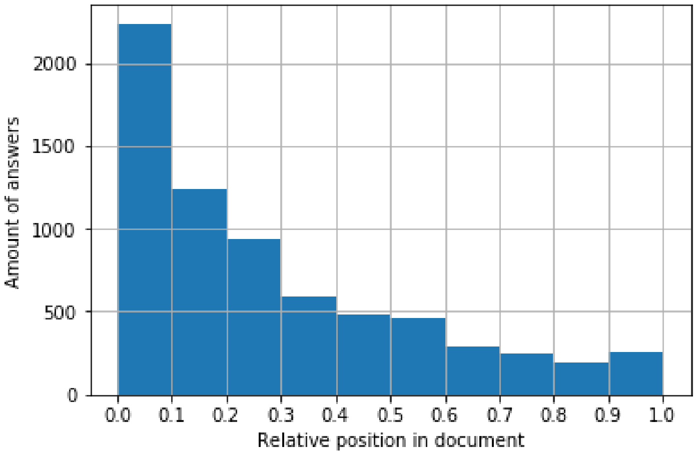


Figure 3.5: Amount of answers (y-axis) per relative position of answer passages (x-axis) among Wikipedia pages split into passages of six sentences: 0-0.10 means the passage was among the first 10% of passages in the document, 0.9-1.0 means the passage was among the last 10% of passages in the document.

3.5. Diagnostic Datasets

In this section we elaborate upon the obtained diagnostic datasets after employing our diagnostic dataset creation pipeline (Figure 3.3) on the passage re-ranking datasets WikiPassageQA [22] and MSMarco [92]. We first show that an axiom conversion was indeed needed to obtain diagnostic datasets across the employed corpora (Section 3.5.1) and then discuss characteristics of the diagnostic datasetswe use in the remainder of this work—specifically the dataset size (3.5.2) and presence of answers (3.5.3).

3.5.1. Are axiom extensions and relaxations necessary for obtaining diagnostic datasets?

For each of the included axioms, we validate that a conversion is indeed required to obtain diagnostic instances across the employed datasets. Considering the original axioms, none of the instances in the WikiPassageQA and only some instances in the MSMarco dataset fulfill their conditions, as can be seen in Table 3.8.

Since WikiPassageQA contains no single-term or two-term queries, we can not find any diagnostic instances for TFC1, TFC2 and M-TDC. Since the dataset also does not contain appended versions of documents also contained in the collection, we can also not find any diagnostic instances for LNC2.

On the other hand, the MSMarco dataset— that is more than two orders of magnitude larger in questions and documents—contains at least one diagnostic instance for three out of the four axioms. However, for TFC1 and TFC2 we can only find a few diagnostic instances, due to the very small amount of one-term queries (3 in the whole dataset) and differences in the length of documents (on average 25.17% of the documents associated with a question are of equal length) in the corpus. For the M-TDC axiom we can however obtain thousands of instances of a query and associated documents. This can partially be explained by the fact that MSMarco contains 1,602 two-term queries.

	TFC1	TFC2	M-TDC	LNC2
WikiPassageQA	0 / 19,044,804	0 / 1,811,123,580	0 / 19,044,804	0 / 19,044,804
MSMarco	32 / 1,396,767,220	1 / 67,773,222,252	7,232 / 1,396,767,220	0 / 1,396,767,220

Table 3.6: Number of instances in the WikiPassageQA and MSMarco corpora that fulfill all conditions per axiom (TFC1, TFC2, M-TDC, LNC2). To put these numbers in perspective, we report the fraction of instances—consisting of a query q and a set of documents d_1, d_2(, d_3 in case of TFC2)—that fulfill all conditions over the total amount of instances (a q with any of such a set of two or three associated documents) present in the pre-processed corpora.

However, one may still question the need for the introduced relaxations that go beyond document length relaxation (subsequent to extension). We incorporated those, as document length relaxation alone was generally insufficient to obtain more than a few diagnostic instances as displayed in Table 3.7. For example, for TFC1 (after extension and only document length relaxation), we only found six instances that could be extracted from WikiPassageQA. This may be explained by the fact that the axiom requires $\boldsymbol{d}_i$ to contain every query term at least once. Similarly, for TFC2, we could only extract 41 instances from MSMarco, which may be due to the precise difference in count for each query term among document triplets as constrained by TFC2. Moreover, the large amount (> 21M) of diagnostic instances we can obtain for M-TDC from MSMarco is misleading since all of these instances (minus the 7K we already obtained as displayed in Table 3.8) consider two documents that contain an equal count for each query term and hence do not allow us to diagnose whether a model adheres to M-TDC (as discussed in Section 3.3.4). Furthermore, note that despite the fact that there exist hundreds of documents that are duplicates of each other in MSMarco (as discussed in Section 3.4.2), we can only obtain one instance that fulfills the conditions of extended version of LNC2 (i.e. a document for which a duplicate exists, while both contain every query term at least once). Concluding, we can only obtain a large amount of instances for the TFC1 axiom after extension and document relaxation.

	TFC1	TFC2	M-TDC	LNC2
WikiPassageQA	6 / 19,044,804	0 / 1,811,123,580	1 / 19,044,804	0 / 19,044,804
MSMarco	2,245,323 / 1,396,767,220	41 / 67,773,222,252	21,625,566 / 1,396,767,220	1 / 1,396,767,220

Table 3.7: Number of instances in the WikiPassageQA and MSMarco corpora that fulfill all conditions per axiom (TFC1, TFC2, M-TDC, LNC2) after *extending* the axiom and applying *document length relaxation*. Similar as before, we report the fraction of instances that fulfill all conditions over the total amount of instances present in the pre-processed corpora.

On a more general level, we conclude that axiom conversions are indeed required to 1) obtain a sufficient amount of diagnostic instances across axioms and datasets and 2) allow to widen the scope of axiomatic analyses to include a range of queries (i.e. not only one- or two-term queries).

3.5.2. How does the size of diagnostic datasets differ across axioms and corpora?

Respectively in Table 3.8 and Table 3.9 we present the number of diagnostic instances we have obtained for the WikiPassageQA and MSMarco datasets per extended, relaxed axiom, following the methodology described in Section 3.2.

Let us first consider the three axioms ($\overline{\texttt{TFC1}}$, TFC2 and $\overline{\texttt{M-TDC}}$) based on data extraction. Depending on the axiom, we have extracted between 42K ($\overline{\texttt{M-TDC}}$) and 3.5M (TFC1) instances for WikiPassageQA and between 25K (TFC2) and 152M (TFC1) instances for MSMarco. From the MSMarco dataset—that is two orders of magnitude larger in questions and documents—we have extracted diagnostic datasets that are roughly two orders of magnitude larger, except for $\overline{\texttt{TFC2}}$. From the large difference in diagnostic instances extracted for $\overline{\texttt{TFC2}}$ from WikiPassageQA (1M) and MSMarco (2.6K), it becomes clear that the amount of diagnostic instances that can be extracted from different corpora for one axiom can differ significantly. They also significantly differ per axiom, as across the datasets we obtain a large amount (millions) of diagnostic instances for TFC1, whereas for TFC2 we obtain a large amount (over a million) diagnostic instances from WikiPassageQA but a very small amount (a few thousand) diagnostic instances from MSMarco.

Let us now consider LNC2, whose instances are not extracted from the corpus, but instead were generated based on the original corpus. We created instances with $k = 2,3,4$ times the original content and maintain the original labels (e.g. a passage that was labeled relevant in its original form is labeled relevant in its artificial form as well, as supported by the $\overline{\texttt{LNC2}}$ axiom). We only considered passages up to 240 words in eventual length (in both corpora), due to experimental constraints[19], leading to a total of 100K instances for the two variants of LNC2 in WikiPassageQA and 50M in MSMarco. The reason for this difference (that is larger than the two order of magnitude difference in the amount of questions in both datasets) lies in the fact that MSMarco contains relatively more documents that are shorter than 240 words (recall Fig. 3.4).

Finally, we note that there is no overlap between the obtained diagnostic instances for our $\overline{\texttt{TFC1}}$, $\overline{\texttt{TFC2}}$, $\overline{\texttt{M-TDC}}$ and $\overline{\texttt{LNC2}}$ axioms due to how the axioms are defined: similar as for the original axioms TFC1, TFC2, M-TDC and LNC2, there is no instance of a query and documents so that the conditions of multiple axioms hold.

	$\overline{\texttt{TFC1}}$	$\overline{\texttt{TFC2}}$	$\overline{\texttt{M-TDC}}$	$\overline{\texttt{LNC2}}$
Parameters				$k = \{2,3,4\}$, $doc\_len_{\max} = 240$
Train	2,758,223	837,838	32,830	82,785
Dev	376,902	50,772	3,837	10,485
Test	353,621	183,898	4,391	10,074
Total	3,488,746	1,072,508	41,058	103,344

Table 3.8: Number of instances per axiom ($\overline{\texttt{TFC1}}$, $\overline{\texttt{TFC2}}$, $\overline{\texttt{M-TDC}}$, $\overline{\texttt{LNC2}}$) per split (train/dev/test) in the WikiPassageQA corpus.

	$\overline{\texttt{TFC1}}$	$\overline{\texttt{TFC2}}$	$\overline{\texttt{M-TDC}}$	$\overline{\texttt{LNC2}}$
Parameters				$k = \{2,3,4\}$, $doc\_len_{\max} = 240$
Train	120,900,840	20,204	3,512,195	40,021,304
Dev	15,525,229	3,013	455,276	5,149,832
Test	15,078,668	2,530	437,225	4,971,208
Total	151,504,737	25,747	4,404,696	50,142,344

Table 3.9: Number of instances per axiom ($\overline{\texttt{TFC1}}$, $\overline{\texttt{TFC2}}$, $\overline{\texttt{M-TDC}}$, $\overline{\texttt{LNC2}}$) per split (train/dev/test) in the MSMarco corpus.

[19] Concretely, when using the MatchZoo toolkit for our neural models we ran into issues when the maximum document length was set to include longer passages, see also https://github.com/faneshion/MatchZoo/issues/264.

3.5.3. To what extent do axioms rank relevant documents above non-relevant documents?

Axioms typically prescribe that a certain document $\boldsymbol{d_i}$ should have an RSV larger than (or equal to) the RSV of another document $\boldsymbol{d_j}$ (for the axioms considered in this work, only TFC2 does not follow this typical form). In Section 2.5 it was introduced that, according to the axiomatic thinking approach to IR, a model that fulfills more axioms tends to achieve a higher retrieval effectiveness. Hence, given these two statements, we would intuitively expect that documents $\boldsymbol{d_i}$ would more often happen to be relevant than documents $\boldsymbol{d_j}$. However, whether or not this holds for a diagnostic instance may differ per dataset. For example, consider a dataset that comprises of queries with many verbose query terms. In that case, the TFC1 axiom—that prescribes that a document that has a larger absolute total count of query terms than another document should have a larger RSV—may very well prescribe to rank a non-relevant document (that contains more of verbose query terms) above a relevant document (that contains less of verbose query terms). Instead, the M-TDC axiom—that prescribes that a model should account for how popular a term is in the corpus—may in that case very well more often rank a relevant document (that contains many non-verbose query terms) above a non-relevant document (that contains just as much verbose query terms) than the other way around.

Note that in this example, we only consider the cases where only one document ($\boldsymbol{d_i}$ or $\boldsymbol{d_j}$) is relevant. However, axiomatic instances can actually consist of four different type of pairs of documents:

- two relevant documents $\boldsymbol{d_i}, \boldsymbol{d_j}$;
- a relevant document $\boldsymbol{d_i}$ and a non-relevant document $\boldsymbol{d_j}$;
- a non-relevant document $\boldsymbol{d_i}$ and a relevant document $\boldsymbol{d_j}$;
- two non-relevant documents $\boldsymbol{d_i}, \boldsymbol{d_j}$;

Hence, due to the small amount of relevant documents (3% of the candidate passages in WikiPassageQA and 2% of the candidate passages in MSMarco), we can expect that a large amount of the diagnostic instances considers pairs of two non-relevant passages and a very small amount considers two relevant passages.

In Table 3.10 and Table 3.11, we display how often each combination of pairs of $\boldsymbol{d_i}$ and $\boldsymbol{d_j}$ is found among the obtained diagnostic instances for respectively the WikiPassageQA and MSMarco dataset. With $\boldsymbol{d_i} > \boldsymbol{d_j}$, we denote that $\boldsymbol{d_i}$ should be ranked higher than $\boldsymbol{d_j}$ (which evidently holds if $\boldsymbol{d_i}$ should receive a higher score than $\boldsymbol{d_j}$). With $\boldsymbol{d_i} \geq \boldsymbol{d_j}$ we denote that $\boldsymbol{d_i}$ should not be ranked lower than $\boldsymbol{d_j}$.

Let us first discuss WikiPassageQA. From these numbers, we can conclude that the sheer amount of diagnostic instances indeed considers two non-relevant passages, as expected. We furthermore can obtain that for both the TFC1 and M-TDC axiom, the instances more often consider a relevant passage $\boldsymbol{d_i}$ and a non-relevant passage $\boldsymbol{d_j}$ than the other way around (3.19% vs 0.44% in TFC1 and 1.21% versus 0.85% in M-TDC). We also see that for TFC1 the difference between both type of pairs $\boldsymbol{d_i}, \boldsymbol{d_j}$ is more than a factor seven, whereas for M-TDC the difference is less than a factor 2. From this, we may hypothesize that the diagnostic instances for the TFC1 axiom would be a better indicator of retrieval effectiveness than the diagnostic instances for M-TDC (for WikiPassageQA). Under that hypothesis, a model that has a better axiomatic performance on TFC1 more likely to have a better retrieval effectiveness for WikiPassageQA than a model that has a better axiomatic performance on M-TDC. We will further discuss this hypothesis in Section 4.2.1.

Now let us discuss the obtained numbers for MSMarco. We again find that the sheer amount of diagnostic instances indeed considers two non-relevant passages. However, we obtain that a significant larger part of the LNC2 instances consider two relevant documents $\boldsymbol{d_i}, \boldsymbol{d_j}$ (7.16%) compared to WikiPassageQA (1.58%). This can be explained by the fact that MSMarco contains *relatively* more answer passages than non-answer passages that are shorter than the adopted maximum length of 240 words compared to WikiPassageQA(recall Fig. 3.4). Moreover, for TFC1 and TFC2 we obtain very different results compared to WikiPassageQA. For MSMarco, for both TFC1 and TFC2 we can find more instances that consider a non-relevant document $\boldsymbol{d_i}$ and a relevant document $\boldsymbol{d_j}$ than the other way around. From this, we may hypothesize that both axioms may not give be good indicators of retrieval performance. We will further research this hypothesis in Section 4.2.2.

Furthermore, in addition to an explanation for the experienced retrieval effectiveness, these numbers may provide us with directions on how we can improve the performance of a model. For instance, we may hypothesize that improving the performance of a model on TFC1 in the WikiPassageQA dataset is more likely to have a positive effect than fixing M-TDC. We will further research this in Chapter 4.4.

$\boldsymbol{d_i \succ d_j}$	$\overline{\texttt{TFC1}}$		$\overline{\texttt{M-TDC}}$		$\overline{\texttt{LNC2}}$	
relevant $\succ$ **relevant**	803	0.02%	6	0.01%	1,685	1,58%
relevant $\succ$ **non-relevant**	111,372	3.19%	526	1.25%	0	0.00%
non-relevant $\succ$ **relevant**	15,336	0.44%	374	0.89%	0	0.00%
non-relevant $\succ$ **non-relevant**	3,362,038	96.37%	41,058	97.84%	104,957	98.42%
total	3,488,746	100.00%	41,964	100.00%	106,642	100.00%

Table 3.10: Presence of relationships as prescribed per axiom for the diagnostic datasetsobtained from the `WikiPassageQA` dataset. Percentages have been rounded to two decimals. Note that for $\overline{\texttt{M-TDC}}$, $\succ$ should be replaced with $\succeq$.

$\boldsymbol{d_i \succ d_j}$	$\overline{\texttt{TFC1}}$		$\overline{\texttt{M-TDC}}$		$\overline{\texttt{LNC2}}$	
relevant $\succ$ **relevant**	15,557	0.01%	602	0.01%	1,442,332	7.16%
relevant $\succ$ **non-relevant**	2,880,652	1.90%	58,648	1.33%	0	0%
non-relevant $\succ$ **relevant**	5,348,529	3.53%	99,825	2.27%	0	0%
non-relevant $\succ$ **non-relevant**	143,259,999	94.56%	4,245,621	96.39%	18,700,012	92.84%
total	151,504,737	100.00%	4,404,696	100.00%	20,142,2344	100.00%

Table 3.11: Presence of relationships as prescribed per axiom for the diagnostic datasets obtained from the `MSMarco` dataset. Percentages have been rounded to two decimals. Note that for $\overline{\texttt{M-TDC}}$, $\succ$ should be replaced with $\succeq$.

4

Diagnosing Neural IR Models

In this chapter we elaborate upon how we have used the previously created diagnostic datasets for diagnosing neural IR models. We discuss the experimental setup that was adopted in this work in Section 4.1, followed by the results of the conducted diagnostic experiments in Section 4.2. Finally, we research the impact of document length differences within diagnostic instances in Section 4.3 and briefly research a methodology aimed at fixing neural IR models in Section 4.4.

4.1. Experimental Setup

We have diagnosed 3 traditional and 6 neural IR models with 9 diagnostic datasets for 4 axioms obtained from 2 original datasets following the methodology discussed in Chapter 3. In Section 4.1.1, we first provide an overview of our methodology of diagnosing retrieval models. Then, in Section 4.1.2, we discuss the models we have selected to diagnose in our experiments and to what extent we have tuned their parameters. Finally, in Section 4.1.3, we discuss the sanity checks we executed to validate the proper functioning of the employed models on benchmarks.

4.1.1. Methodology

An overview of how we obtain the axiomatic performance and retrieval effectiveness of models is depicted as a pipeline in Fig. 4.1, which is further explained in this section. After pre-processing the queries and documents in an original dataset, like `WikiPassageQA` and `MSMarco`, we feed instances of a query and associated documents to (trained Neural) IR models for scoring. Each model then outputs a list of ranked documents for each query. We can evaluate the output of a model based upon the ranked documents and relevance labels per query, resulting in a measure of retrieval effectiveness. Next to this evaluation, we also conduct a diagnosis of a model based upon the ranked documents per query and diagnostic instances for that query, resulting in a measure of axiomatic performance.

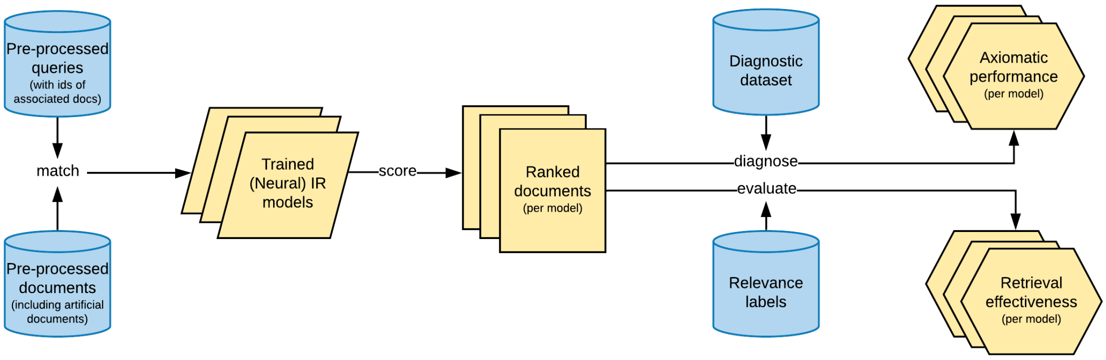


Figure 4.1: Overview of using a diagnostic dataset to diagnose IR models: given a pre-processed dataset (including artificial data, in our case required for $\overline{\texttt{LNC2}}$) and a diagnostic dataset, we can obtain the axiomatic performance and retrieval effectiveness of a model [1].

[1] For completeness, we note that we have not modelled model-specific components such as pseudo-relevance feedback and query expansion (employed in RM3) and matching histograms (employed in DRMM) in this figure.

While the majority of this pipeline follows the traditional Cranfield methodology (introduced in Section 2.1.3), it differs in two parts. First of all, we introduce the component that employs diagnostic datasets to obtain axiomatic performances (the "diagnose" part in the top right in Fig. 4.1), whereas the Cranfield-style experiments only consider evaluation (the "evaluate" part in the bottom right in Fig. 4.1). Secondly, we sometimes enrich original datasets with artificial documents (bottom left in Fig. 4.1) so that we can obtain the axiomatic performance for axioms for which there are little to no diagnostic instances, as discussed in Section 3.5.1. We note that the evaluations on such artificial data require extra effort: regarding the *neural models* we propose to enrich the input data with the artificial data and regarding the *traditional models* we propose to rank artificial documents on a per-document basis. Both approaches are not displayed in the abstract overview displayed in Fig. 4.1, but are further detailed in the next paragraph.

Diagnosis on artificial data

Neural models are trained on a split of a dataset, validated on another and subsequently tested on a final split. To obtain decent performance on this test, the models typically need to be trained (and validated) on instances that come from the same distribution as the test split. We therefore propose two strategies for diagnosing neural models on artificial data: 1) we only add artificial data to the test split and 2) we add artificial data to all splits.

Recall that the artificial datasets on which we diagnose neural models were added to the original datasets because the original datasets did not have sufficient instances that fulfill the conditions of an axiom. Hence, if we only add artificial documents to the test set (under the first strategy), this is likely to impact the performance of neural models on this test set as the test split including artificial documents follows a different distribution than the train and dev split that only include documents in the original corpus. Hence, it may be more "fair" to also incorporate the artificial documents in the training and development splits of the dataset (the second strategy). In our experiments we have researched the results of both strategies, to which we respectively refer as $\overline{\texttt{AXIOM}}^{Test}$ and $\overline{\texttt{AXIOM}}^{All}$.

Traditional models maintain an index of documents in the document collection and utilize index-based statistics to rank documents according to their retrieval formula. The addition of artificial documents to this index impacts such statistics (such as document frequency), and can therefore impact the document ranking returned by retrieval models. To minimize such potential impact, we propose to rank artificial documents on a per-document basis: we first add one document to the corpus, then let a model rank all queries to which this document is associated and after storing the resulting ranking, we remove the document from the index, so that we once again obtain the original index.

For these models, we hence obtain the same result for $\overline{\texttt{AXIOM}}^{Test}$ as $\overline{\texttt{AXIOM}}^{All}$, since the traditional models are now—unlike the neural models that are trained—*not* influenced by the presence or absence of artificial documents in the train and dev split.

4.1.2. Retrieval model selection and configuration

In the following, we subsequently introduce the traditional and neural IR models we have employed in this work, we elaborate upon why they were included as well as their tuning process and adopted configurations. Due to experimental issues that rise when running neural retrieval models on the *whole* `MSMarco` dataset—containing over 500K queries and over 5 million documents after pre-processing as discussed in Section 3.4.2)—we use a **random**[2] **10% subset of the questions in the** `MSMarco` dataset in the experiments discussed throughout this chapter. We have only kept the documents associated with these questions (i.e. the documents that have been returned by our QL pre-ranker for these questions) in our corpus. We will henceforth in this chapter refer to this subset when stating "`MSMarco`" unless explicitly stated otherwise.

Traditional IR models

For our experiments we employed the open-source search engine `Indri`, available with the Lemur toolkit[3] [119]. Indri has been used by IR researchers for over a decade (e.g. [36, 109, 132] and neural IR papers such as [3, 60, 87]) and is compatible with various operating systems. We included the widely adopted traditional retrieval baselines Okapi BM25 and query likelihood with Dirichlet smoothing, but also included the generally stronger [75, 84, 100] RM3 model which encompasses a query expansion component.

We tuned the parameters of BM25, QL and RM3 on the combined train and development parts of `WikiPassageQA`, optimizing for MAP, and on the train and development parts of `MSMarco`, optimizing for MRR. An overview of the tested parameters can be found in Appendix A. The best-performing parameter settings have been adopted in the experiments in this work and can be found in Table 4.1.

Model	**Parameter**	**Adopted value per dataset**	
		`WikiPassageQA`	`MSMarco`
BM25	$k1$	0.4	0.6
	b	0.1	1.25
	$k3$	1.0	1.0
QL	μ	750.0	10.0
RM3	μ	750.0	10.0
	$fbDocs$	5.0	20.0
	$fbTerms$	500.0	100.0
	$fbMu$	3000.0	3000.0
	$fbOrigWeight$	0.6	0.6

Table 4.1: Adopted values per parameter per traditional model employed in this work for the test splits of the `WikiPassageQA` and `MSMarco` datasets after tuning on the train and dev splits.

As introduced, we adopted a per-document scoring scheme for diagnosing traditional models on the $\overline{\texttt{LNC2}}$ axiom. This has resulted in large computational requirements, especially for obtaining the per-document scores for the $\overline{\texttt{LNC2}}$ axiom per traditional model on the `MSMarco` dataset (for which we have nearly 500K appended versions of documents smaller than the limit of 240 words in the test split of the subset). Hence, we have tested the traditional models on a random[4] subset of 5% of the diagnostic instances in the test split of `MSMarco`, i.e. 0.5% of the *original* `MSMarco` dataset. We hence consider 24,583 diagnostic instances obtained from 277 questions and their associated 12,420 documents.

[2] We employ `random.seed(2018)`.

[3] See `https://www.lemurproject.org/indri.php`, we have employed Indri 5.13.

[4] Since we have obtained a large amount of diagnostic instances for $\overline{\texttt{LNC2}}$ for nearly all questions in `MSMarco` we do not need to consider a specific subset to maintain sufficient diagnostic instances and hence take a random subset, see also [2].

Neural IR models

For our neural models, we employed the MatchZoo neural retrieval toolkit[5] [38]. The toolkit consists of more than 10 (representation-based, interaction-based and hybrid) neural model implementations and has been employed in a number of prior studies, including [15, 103, 135][6].

Initially, we considered all neural models implemented in MatchZoo; however, for a number of models we observed a significant drop in retrieval effectiveness when compared with published benchmarks, as will be further discussed in Section 4.1.3. We have therefore turned our attention to the 6 best-performing models, most of which are interaction-based (in line with the findings reported in [93]). Since all models have been detailed in Section 2.3.3, we here only provide a brief recap of each:

- Arc-I [58], a siamese network that separately summarizes the meaning of two sentences through one-dimensional layers of convolution and pooling and finally matches them with an MLP;
- MatchPyramid [96], a symmetric, interaction-based model that employs convolutional neural networks in a two-dimensional manner, mimicking image recognition in its text matching;
- MV-LSTM [126], a symmetric, interaction-based model that generates an interaction matrix using a bi-directional LSTM that aims to capture a representation of the context rather than separate words;
- Duet [87], a hybrid of an interaction-based and representation-based model: it combines two separate deep nets, one aimed at semantic matching, and another aimed at positional matching;
- DRMM [48], an asymmetric, interaction-based model that employs a histogram representation and a term gating network to determine the similarity between a query and a document;
- aNMM [133], an asymmetric, interaction-based model that employs a value-shared weighting scheme and a question attention network.

MatchZoo contains architecture configurations[7] that have been optimized for the WikiQA dataset [134], an open-domain question answering dataset, similar in spirit to WikiPassageQA, though defined on the document, not passage level. Since neural architecture search [143] is beyond the scope of this work, we maintained the default MatchZoo configurations, including random seeds[8] as well as learning rates and optimizers as optimized for the WikiQA dataset. More details on the configuration of the individual models can be found in the configuration files[9]. Due to the computational requirements of neural model training, we limited the maximum query length and passage length for both datasets. For WikiPassageQA we limit them to 20 and 240 terms respectively and for MSMarco we limit them to 30 and 289—these settings meant that in more than 99% of all instances per dataset the entire question and entire passage was considered. All neural models were trained for 400 iterations.

As introduced, we adopted two strategies for diagnosis with regard to the $\overline{\texttt{LNC2}}$ axiom: 1) we simply tested a model trained on the regular corpus on our diagnostic dataset for $\overline{\texttt{LNC2}}$ including artificial instances ($\overline{\texttt{LNC2}}^{Test}$) and 2) we trained a model on the regular corpus combined with artificial instances (maintaining a training scheme of 400 iterations - hence these models have been trained on less instances from the regular corpus) and tested it on our diagnostic dataset ($\overline{\texttt{LNC2}}^{All}$).

[5] See https://github.com/NTMC-Community/MatchZoo, we have employed MatchZoo as obtained from commit e564565.

[6] At the time of writing the repository has over 2000 stars and nearly 600 forks.

[7] See https://github.com/faneshion/MatchZoo/tree/e564565/examples/wikiqa/config for the configurations (e.g. learning rate, optimizer, layers) per model.

[8] MatchZoo employs random.seed(49999), numpy.random.seed(49999) and tensorflow.set_random_seed(49999).

[9] See https://github.com/NTMC-Community/MatchZoo/tree/e564565/examples/wikiqa/config.

4.1.3. Sanity checks

We adopted sanity checks to validate whether the models we would like to employ in this research, work as expected. We compare the results we obtain with traditional models (BM25 and QL[10]) and 11 neural models with benchmark scores on 2 datasets (`WikiQA` and `WikiPassageQA`[11]). As a result of these sanity checks we have filtered out 5 of the 11 neural models in the other experiments conducted in this thesis.

Validation of neural models on the `WikiQA` dataset

We have run 11 neural models in MatchZoo on the `WikiQA` dataset [134] for which all required code is provided in the MatchZoo repository. Table 4.2 displays the results as provided by [38] (left) and the results of our run on MatchZoo (right). We obtain deviations ranging from 0.0021 to 0.3448 per metric. These deviations may for example originate from a difference in the use of different hardware and threading (and their interactions) [26]. However, we consider the scores that deviate more than 0.1 from the reported benchmark values to be indicators that the models may not work properly. Accordingly, we consider scores that deviate less than 0.1 to be well within expected differences and indicators that the models are working properly.

Validation of BM25 and QL on the `WikiPassageQA` dataset

We have run 2 traditional retrieval models (BM25 and QL) in Indri on the `WikiPassageQA` dataset [22], pre-processed as discussed in Section 3.4.1. Table 4.3 displays the reported retrieval effectiveness in the benchmark (upper rows) and the retrieval effectiveness we achieved with our BM25 and QL (lower rows). We can see minor deviations ranging from 0.0004 up to 0.0368 in BM25 and from 0.0025 up to 0.0192 in QL across the four metrics. These differences may be the result of different pre-processing (e.g. we have not removed any stop words). All in all, we conclude that the traditional IR models perform as expected.

Validation of neural models on the `WikiPassageQA` dataset

We have run 11 neural models in MatchZoo on the `WikiPassageQA` dataset [22], pre-processed as discussed in Section 3.4.1. Table 4.4 displays the reported retrieval effectiveness in the benchmark (which includes different models) and the retrieval effectiveness of our employed neural models. As in [22], only a few neural models outperform the baselines (presented in Table 4.3). Moreover, we can find that the interaction-based models (with the exception of ARC-II) outperform the representation-based models.

For a number of models (especially the representation-based models such as CDSSM) we observe a significant drop in retrieval effectiveness in the `WikiPassageQA` dataset compared to `WikiQA`. This lack of model robustness to the corpus is a well-known problem for neural models. Due to both their large deviation from the benchmark (as obtained in the validation of neural models on the `WikiQA` dataset and their low-performance across two datasets (`WikiQA` and `WikiPassageQA`), we exclude the K-NRM, CDSSM and ARC-II models in the remainder of our research. We however keep the ARC-I, MV-LSTM and Duet models that perform well below (i.e. lower than 0.25 in MAP) other neural models in the benchmark (that score higher than 0.33 in MAP) and the adopted baselines (that score higher than 0.53 in MAP) as they did achieve reasonable performance on the `WikiQA` dataset with a small deviation from the provided benchmark.

[10] We do not include the RM3 retrieval model which we employ in this work in these sanity checks as we have not found reported retrieval effectiveness of this model on any of the employed datasets.

[11] For the `MSMarco` dataset there are only leaderboard scores available which were obtained with an held-out test set and under different experimental setups compared to our setup (e.g. answer passages are not added to the candidate documents if they are missing in the benchmark). Hence, we do not report a side by side comparison of the scores of our neural and traditional retrieval models and the reported values in the leaderboard.

	Benchmark		MatchZoo	
	nDCG@5	MAP	nDCG@5 (deviation)	MAP (deviation)
ARC-I	0.6317	0.5870	0.6356 (+0.0039)	0.5849 (+0.0021)
DSSM	0.6134	0.5647	0.6088 (−0.0046)	0.5681 (+0.0034)
MatchPyramid	0.6913	0.6434	0.6816 (−0.0097)	0.6404 (−0.0030)
DUET	0.6722	0.6301	0.6609 (+0.0113)	0.6236 (−0.0065)
MV-LSTM	0.6452	0.5988	0.6593 (−0.0141)	0.6217 (+0.0229)
DRMM-TKS	0.6956	0.6586	0.6801 (−0.0155)	0.6364 (−0.0222)
aNMM	0.6696	0.6297	0.6435 (−0.0261)	0.6063 (−0.0234)
DRMM	0.6621	0.6195	0.6103 (−0.0518)	0.6498 (+0.0510)
K-NRM	0.6693	0.6256	0.5341 (−0.1352)	0.4929 (−0.1327)
CDSSM	0.6084	0.5593	0.4398 (−0.1686)	0.4045 (−0.1548)
ARC-II	0.6176	0.5845	0.2728 (−0.3448)	0.2628 (−0.3217)

Table 4.2: Reproduction of the benchmark on the WikiQA dataset provided by [38] (left) with our run of MatchZoo (right) sorted on their difference on both metrics (deviation) which is considerably large (>0.1) for K-NRM, CDSSM and Arc-II.

		MAP (deviation)	**MRR** (deviation)	**P@5** (deviation)	**P@10** (deviation)
Benchmark	BM25	0.5373	0.6258	0.1947	0.1151
	QL	0.5436	0.6338	0.1947	0.1151
Indri	BM25	0.5199 (−0.0174)	0.5983 (−0.0275)	0.1821 (−0.0126)	0.1155 (+0.0004)
	QL	0.5355 (−0.0081)	0.6209 (−0.0129)	0.1913 (−0.0034)	0.1176 (+0.0025)
		nDCG (deviation)	**Rec.@5** (deviation)	**Rec.@10** (deviation)	**Rec.@20** (deviation)
Benchmark	BM25	0.6659	0.6334	0.7311	0.8309
	QL	0.6715	0.6353	0.7275	0.8426
Indri	BM25	0.6513 (−0.0146)	0.5966 (−0.0368)	0.7343 (+0.0032)	0.8262 (−0.0047)
	QL	0.6653 (−0.0062)	0.6227 (−0.0126)	0.7467 (+0.0192)	0.8396 (−0.0030)

Table 4.3: Reproduction of the benchmark on the WikiPassageQA dataset as provided by [22] (upper rows) with our run in Indri (lower rows) per metric. We consider these deviations (<0.04) to be small.

		MAP	MRR	P@5	P@10	nDCG	Recall@5	Recall@10	Recall@20
Benchmark	LSTM	0.3352	0.3947	0.1197	0.0780	0.4912	0.3915	0.5894	0.7169
	CNN+TF	0.4009	0.4581	0.1572	0.1099	0.5577	0.5212	0.7024	0.8412
	LSTM-CNN+TF	0.3577	0.4156	0.1351	0.0942	0.5196	0.4538	0.6187	0.7608
	C+WCNN-LSTM	0.4385	0.5534	0.1728	0.1104	0.5837	0.5709	0.6931	0.8326
	M-CNN-LSTM+TF	**0.5608**	**0.6792**	**0.2083**	**0.1228**	**0.6791**	**0.6522**	**0.7329**	**0.8592**
MatchZoo	CDSSM	0.1801	0.2088	0.0633	0.0546	0.3642	0.2137	0.3524	0.5635
	ARC-I	0.1950	0.2226	0.0739	0.0621	0.3800	0.2533	0.4099	0.6378
	ARC-II	0.1986	0.2239	0.0700	0.063	0.3828	0.2424	0.4288	0.6529
	K-NRM	0.2255	0.2661	0.0918	0.0742	0.4103	0.3040	0.4776	0.6813
	MV-LSTM	0.2337	0.2678	0.0903	0.0790	0.4171	0.3090	0.5085	0.7496
	Duet	0.2472	0.2877	0.0971	0.0780	0.4277	0.3219	0.4981	0.7039
	MatchPyramid	0.4388	0.5088	0.1807	0.1147	0.5908	0.5917	0.7233	0.8539
	DRMM-TKS	0.5398	0.6183	0.1966	0.1229	0.6697	0.6484	0.7779	0.8657
	DRMM	0.5597	0.6393	0.2024	0.1213	0.6841	0.6554	0.7674	0.8642
	aNMM	**0.5734**	**0.6579**	**0.2087**	**0.1263**	**0.6974**	**0.6781**	**0.7963**	**0.8857**

Table 4.4: Comparison of deep nets adopted in the WikiPassageQA benchmark as provided by [22] (top) and the deep nets we ran in MatchZoo (bottom). In both sets only few models outperform the baselines (presented in Table 4.3) as represented by the dashed line.

4.2. Results

In this section we will discuss the results of our experiments. In Section 4.2.1 we discuss the results of our experiments on `WikiPassageQA`, followed by the results of our experiments on `MSMarco` in Section 4.2.2. All measures of retrieval effectiveness have been obtained using `trec_eval`[12]. Moreover, all statistical significance measurements on *retrieval effectiveness* have been conducted with the Wilcoxon test while discarding all values in which there is no difference between two samples[13], whereas significance measurements on *axiomatic performance* have been conducted with the McNemar test[14].

4.2.1. Diagnostic experiments on `WikiPassageQA`

Table 4.6 displays both the retrieval effectiveness measured on the original corpora, as well as the axiomatic performance measured on our diagnostic datasets for `WikiPassageQA`. An overview with more detailed numbers can be found in Appendix B.1.1.

	Retrieval effectiveness			**Performance per axiom**				
	MAP	MRR	P@5	$\overline{\texttt{TFC1}}$	$\overline{\texttt{TFC2}}$	$\overline{\texttt{M-TDC}}$	$\overline{\texttt{LNC2}}^{Test}$	$\overline{\texttt{LNC2}}^{All}$
Random				0.50		0.50	0.50	0.50
^{1} BM25	0.52^{4,5,6,7}	0.60^{4,5,6,7}	0.18^{4,5,6}	0.73	**0.98**	**1.00**	**0.80**	**0.80**
^{2} RM3	0.53^{1,4,5,6,7}	0.62^{1,4,5,6,7}	**0.19**^{1,4,5,6}	**0.88**	0.63	0.94	0.72	0.72
^{3} QL	**0.54**^{1,4,5,6,7}	**0.62**^{1,4,5,6,7}	**0.19**^{1,4,5,6}	0.87	0.63	0.94	0.68	0.68
^{4} Arc-I	0.20	0.22	0.07	0.68	0.55	0.50	0.13	0.39
^{5} MV-LSTM	0.23^{4}	0.27^{4}	0.09^{4}	0.68	0.56	0.51	0.16	**0.71**
^{6} Duet	0.25^{4}	0.29^{4}	0.10^{4}	0.69	0.56	0.48	0.19	0.47
^{7} MatchP.	0.44^{4,5,6}	0.51^{4,5,6}	0.18^{4,5,6}	0.79	0.58	0.63	0.00	0.19
^{8} DRMM	0.56^{1,2,4,5,6}	0.64^{1,2,3,4,5,6}	0.20^{1,2,3,4,5,6}	0.84	**0.60**	**0.76**	0.05	0.12
^{9} aNMM	**0.57**^{1,2,3,4,5,6,7}	**0.66**^{1,2,3,4,5,6,7}	**0.21**^{1,2,3,4,5,6,7}	**0.85**	0.56	0.69	**0.38**	0.47
Best in [22]	0.56	0.68	0.21	?	?	?	?	?

Table 4.5: Overview of models' retrieval effectiveness and fraction of fulfilled axiom instances. For measuring statistical significance, we employed the Wilcoxon test and McNemar test with $p < 0.05$ on respectively measures for retrieval effectiveness and axiomatic performance (regarding the latter all scores are significantly different in $\overline{\texttt{TFC1}}$, all excluding RM3 and QL as well as ARC-I and MV-LSTM in $\overline{\texttt{M-TDC}}$ and all excluding MV-LSTM and aNMM in $\overline{\texttt{TFC2}}$).

Retrieval effectiveness

Let us first consider the retrieval effectiveness of our models. As found in several prior studies [48, 97, 99], and as already indicated in [22] with regard to the `WikiPassageQA` dataset, neural models struggle to outperform traditional retrieval baselines that contain just a handful of hyper-parameters. Only DRMM and aNMM are able to significantly outperform the traditional models, with an increase in MAP from 0.54 (QL) to 0.55 (DRMM) and 0.57 (aNMM) respectively. These results are not unexpected, as DRMM is considered to be one of the most competitive neural IR models to date [95], and DRMM and aNMM are similar in the sense that they both employ a specific component to valuate the importance of query terms. Furthermore, similar to [99] we find that DRMM outperforms MatchPyramid, and similar to [60] we observe that MatchPyramid in turn outperforms Duet. Moreover, similar to [126] we find that MV-LSTM outperforms ARC-I and similar to [97, 98] we find that ARC-I shows inferior performance to all other neural and non-neural models under study. Regarding the baselines, we however find that the typically stronger RM3 baseline [100] that employs relevance feedback does not outperform the QL model (although their difference is not significant). One reason for this may be query drift, i.e. a change in the underlying "intent" between the original query and its expanded form [142][15].

[12] See https://trec.nist.gov/trec_eval/, we employed version 8.1.

[13] We have used `scipy.stats.wilcoxon`.

[14] We have used `statsmodels.stats.contingency_tables.mcnemar`.

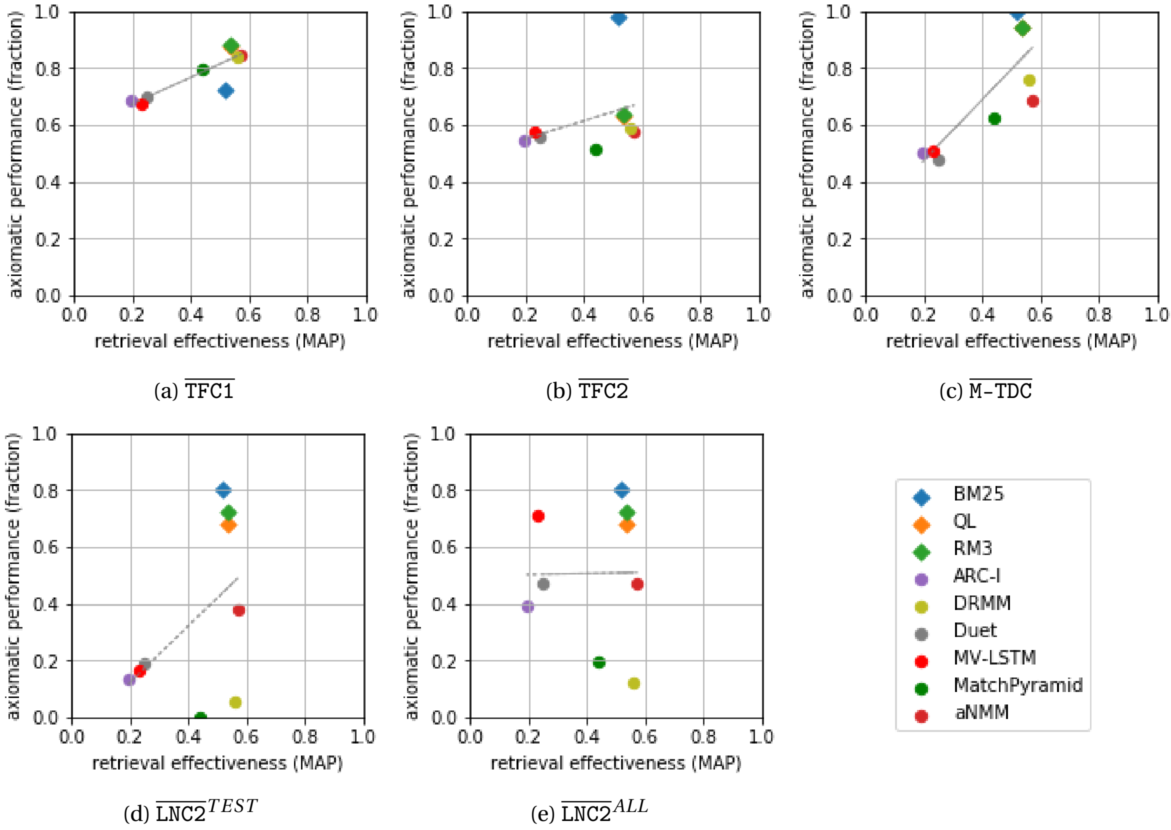
[15] For completeness we note that the RM3 model (MAP=0.5455) does outperform the QL model (MAP=0.5432) on the training and development set under the obtained tuned parameter settings, although not significantly.

Axiomatic performance

Moving on to the axiomatic performance of our models, we find all non-neural models to satisfy the precedence constraints of the vast majority of instances across all four axioms: BM25, RM3 and QL satisfy more than 90% of the $\overline{\texttt{M-TDC}}$ instances and more than 70% of the $\overline{\texttt{TFC1}}$ instances. The largest difference in percentage of satisfied axiomatic instances can be found in $\overline{\texttt{TFC2}}$ (BM25 satisfies 98% of instances, QL only 63%), which can explained by the fact that QL with Dirichlet smoothing employs a document length dependent smoothing component (i.e. longer documents receive less smoothing). Overall, the results are in line with our expectations: as QL and BM25 (sometimes conditionally) satisfy all axioms according to their analytical analyses [42, 43] they should satisfy a large percentage of our extended and relaxed axiomatic instances as well. However, these numbers do not reflect the (un)conditional fulfillment of BM25 and QL per original axiom on a one-to-one basis, for which at least a partial explanation is our relaxation of the document length difference δ, as will further researched in Section 4.3.1.

When we consider the axiomatic scores of our evaluated *neural* models we observe a clear gap: while for $\overline{\texttt{TFC1}}$ (i.e., documents with more query terms should have higher retrieval scores) between 69-85% instances are satisfied, for $\overline{\texttt{TFC2}}$ (i.e., the increase in retrieval score becomes smaller as the absolute term count increases) and $\overline{\texttt{M-TDC}}$ (i.e., documents with more occurrences of rare query terms are favoured) this drops to at most 76%. We can furthermore find that DRMM and aNMM outperform the other neural models by a margin for the $\overline{\texttt{TFC1}}$ and especially the $\overline{\texttt{M-TDC}}$ axiom. Comparing their architectures to the other neural models we study here, we can find that they represent query terms in a more separate manner throughout their architecture (i.e. through separate histogram mappings per query term and separate value-shared weights per query term respectively), which may enable models to better match and aggregate scores from individual query terms (i.e. perform better at $\overline{\texttt{TFC1}}$). Moreover, both models employ a specific component to valuate the importance of query terms (a term-gating network and a question attention network resp.), which may have resulted in a better performance on $\overline{\texttt{M-TDC}}$. Moving on to the $\overline{\texttt{LNC2}}$ axiom, we find that only aNMM is able to learn the underlying pattern to some degree (38% of satisfied instances) without observing instances of duplicated documents in training ($\overline{\texttt{LNC2}}^{Test}$); the remaining neural models correctly rank between 0 and 19% of instances. Once we include the diagnostic dataset instances in the training regime ($\overline{\texttt{LNC2}}^{All}$) all models have learned to some degree that duplicated document content should not be penalized, but still, none of the models is able to satisfy even half of the diagnostic instances. Finally, we note that aNMM achieves a higher retrieval effectiveness than QL and RM3, while QL and RM3 outperform aNMM across all four diagnostic datasets. This is an indication that fulfillment of those four axioms alone is not a perfect indicator of retrieval effectiveness—after all, more than twenty have been proposed in the literature. We leave the evaluation of additional axioms to future work as will be further discussed in Section 5.2.

Now let us consider the correlation between retrieval effectiveness and axiomatic performances. Overall, the correlation between retrieval effectiveness in MAP and the average axiomatic score across all axioms is 0.48 ($N = 9$ retrieval models); this is a positive trend, but not a significant one due to the overall low number of models compared. However, whether axiomatic performance does or does not provide a good diagnosis for retrieval effectiveness, differs per axiom. In Fig. 4.2 we display, for each axiom, the axiomatic performances and retrieval effectiveness (in MAP) per model as obtained from Table 4.5 (note that we could employ a different metric than MAP, such as MRR or P5, but the results would not differ much as can be obtained from Table 4.5). We can see that the diagnostic instances for $\overline{\texttt{TFC1}}$ and $\overline{\texttt{M-TDC}}$ seem to be proper diagnostics: they show a positive relation between axiomatic performance and retrieval effectiveness and all models are relatively close to the trendline that displays this relation (i.e. the Euclidean distance measured as the shortest line perpendicular to the trendline that goes through a point (a model in our figures) [10] is relatively small for each point). $\overline{\texttt{TFC2}}$ also shows a positive relation between axiomatic performance and retrieval effectiveness, but if we would exclude the measurement for BM25, this trend would hardly be positive. For the $\overline{\texttt{LNC2}}$ axioms we obtain that the points are very spread, which we interpret as an indication that these axioms are not a good diagnostic for retrieval effectiveness in the case of `WikiPassageQA`. This becomes even more clear if we exclude the traditional models, in that case we would not obtain a very positive relation for $\overline{\texttt{LNC2}}^{TEST}$ and even a *negative* relation for $\overline{\texttt{LNC2}}^{ALL}$.



(a) $\overline{\texttt{TFC1}}$

(b) $\overline{\texttt{TFC2}}$

(c) $\overline{\texttt{M-TDC}}$

(d) $\overline{\texttt{LNC2}}^{TEST}$

(e) $\overline{\texttt{LNC2}}^{ALL}$

Figure 4.2: Axiomatic performance (fraction of fulfilled diagnostic instances) versus retrieval effectiveness (MAP) of traditional (♦) and neural (•) IR models per axiom for the `WikiPassageQA` dataset. Figures also include a trendline[16]. Note that when QL (displayed in orange) is not clearly visible due to occlusion, it is located behind RM3 (displayed in green).

[16] Obtained with `numpy.polyfit`.

4.2.2. Diagnostic experiments on MSMarco

Table 4.6 displays both the retrieval effectiveness measured on the original corpora, as well as the axiomatic performance measured on our diagnostic datasets for MSMarco[17]. An overview with more detailed numbers can be found in Appendix B.1.2.

	Retrieval effectiveness			Performance per axiom			
	MAP	MRR	P@5	$\overline{\texttt{TFC1}}$	$\overline{\texttt{TFC2}}$	$\overline{\texttt{M-TDC}}$	$\overline{\texttt{LNC2}}^{Test}$
Random				0.50		0.50	0.50
^{1} BM25	0.20	0.20	0.06	0.54	**0.50**	**0.69**	0.00*
^{2} QL	0.21^{1}	0.21^{1}	0.06^{1}	**0.68**	0.43	0.61	0.06*
^{3} RM3	**0.21**^{1,2}	**0.22**^{1,2}	**0.06**^{1,2}	0.66	0.42	0.59	0.00*
^{4} MV-LSTM	$0.25^{1,2,3}$	$0.25^{1,2,3}$	$0.07^{1,2,3}$	0.29	**0.56**	**0.46**	**0.31**
^{5} Arc-I	$0.26^{1,2,3,4}$	$0.26^{1,2,3,4}$	$0.07^{1,2,3}$	0.29	0.47	0.46	0.00
^{6} Duet	$0.28^{1,2,3,4,5}$	$0.29^{1,2,3,4,5}$	$0.08^{1,2,3,4,5}$	0.33	0.42	0.45	0.00
^{7} DRMM	$0.32^{1,2,3,4,5,6}$	$0.32^{1,2,3,4,5,6}$	$0.09^{1,2,3,4,5}$	0.20	0.47	0.38	0.00
^{8} aNMM	$0.32^{1,2,3,4,5,6,7}$	$0.32^{1,2,3,4,5,6}$	$0.09^{1,2,3,4,5,6}$	0.20	0.56	0.39	0.00
^{9} MatchPyramid	**0.33**^{1,2,3,4,5,6,7,8}	**0.34**^{1,2,3,4,5,6,7,8}	**0.09**^{1,2,3,4,5,6,7,8}	**0.35**	0.47	0.46	0.00

Table 4.6: Overview of models' retrieval effectiveness and fraction of fulfilled axiom instances for the MSMarco dataset. For measuring statistical significance, we employed the Wilcoxon test and McNemar test with $p < 0.05$ on respectively measures for retrieval effectiveness and axiomatic performance. Regarding the latter all scores are significantly different in $\overline{\texttt{TFC1}}$ and $\overline{\texttt{M-TDC}}$ except for Duet and MatchPyramid in $\overline{\texttt{M-TDC}}$ and only 12/36 performances were significant for $\overline{\texttt{TFC2}}$: BM25 and aNMM are significantly better than QL, RM3 and Duet, aNMM is also significantly better than DRMM and MV-LSTM is significantly better than QL, RM3, ARCI, Duet and DRMM.

Retrieval effectiveness

Considering the retrieval effectiveness of the models on MSMarco, we obtain very different findings than for WikiPassageQA. We here find that most of the neural models outperform the baselines for the MSMarco dataset: only Arc-I and MV-LSTM do not beat all non-neural baselines (they score up to 0.04 lower in MAP). However, if we only consider the weaker BM25 and QL baselines, we find that all neural models except ARC-I outperform the weak baselines - a finding that has not been obtained in many related works. We conclude that for the MSMarco dataset the traditional IR models struggle, whereas the neural models flourish.

Different from the WikiPassageQA dataset, we now also find that MatchPyramid outperforms all other models, and the non-neural baselines even by a large margin (> 0.10 in MAP). Some other works have found that MatchPyramid outperformed other neural approaches [96, 98] and weak baselines (i.e. a model always returning `true` or the TF-IDF model) [96], but to the best of our knowledge, we are the first to find that MatchPyramid outperforms strong baselines. This can however be explained by the fact that MSMarco may require to use both exact (i.e. it considers questions) and similarity matching signals (i.e. it considers ad-hoc retrieval), while MatchPyramid was designed to capture both signals and treat them as equally important.

Apart from the shift of MatchPyramid, the relative order of neural models based upon their retrieval effectiveness has remained the same as we found for WikiPassageQA - and as discussed in Section 4.2.1, comparable results have been obtained in related works. We furthermore find that the RM3 model now, as we expected, outperforms the QL model by a small, but significant, margin.

[17] Due to experimental limitations, we have not been able to run the $\overline{\texttt{LNC2}}$ experiments for MSMarco, see also [19].

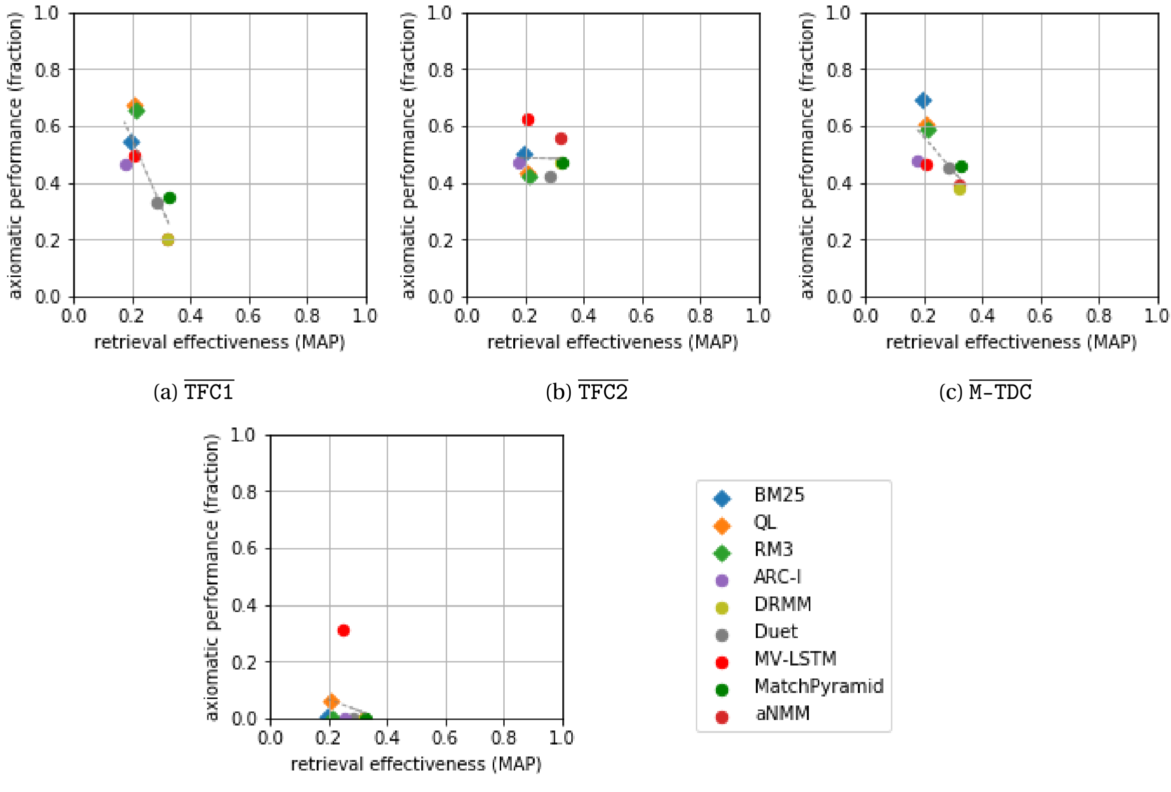
Axiomatic performance

Moving on to the axiomatic performance of our models, we again obtain very different results from our experiments on WikiPassageQA. In our experiment on MSMarco, all non-neural models do satisfy the precedence constraints of the majority of instances across three out of four axioms, although to a lesser extent than in the previous experiment: BM25, RM3 and QL satisfy more than 58% of the $\overline{\texttt{M-TDC}}$ instances and more than 54% of the $\overline{\texttt{TFC1}}$ instances (versus more than 90% of the $\overline{\texttt{M-TDC}}$ instances and 73% of the $\overline{\texttt{TFC1}}$ instances in WikiPassageQA). This difference may stem from the fact that the instances we obtained from MSMarco more often have conditions under which these models do not fulfill the axioms, as well as a different distribution of document length differences δ among documents in a diagnostic instance. We assume this to especially be the case for the $\overline{\texttt{LNC2}}$ axiom for which all traditional models fulfill hardly any diagnostic instances.

When we consider the axiomatic scores of our evaluated *neural* models we observe clear differences, most notably for $\overline{\texttt{TFC1}}$ between 20-35% instances are satisfied and for $\overline{\texttt{LNC2}}$ we obtain that nearly all models fulfill hardly any (roughly 0%). Moreover, the aNMM model that satisfied 38% of the $\overline{\texttt{LNC2}}^{Test}$ instances for WikiPassageQA, now satisfies a rounded 0% of these instances obtained with MSMarco. However, the one model that does fulfill nearly one third (31%) of the instances for $\overline{\texttt{LNC2}}^{Test}$—MV-LSTM—also outperformed all other neural models (and performed on par with traditional models) on $\overline{\texttt{LNC2}}^{All}$ for WikiPassageQA. One possible explanation for this is that the positional representation of MV-LSTM, which is very different from other neural approaches, positively influences the score of the longer, appended versions of documents instead of over-penalizing them.

Now let us consider the correlation between retrieval effectiveness and axiomatic performances. Overall, the correlation between retrieval effectiveness in MAP and the average axiomatic score across all axioms is -0.36 ($N = 9$ retrieval models); this is a negative trend, but not a significant one due to the overall low number of models compared. However, it evidently shows that the overall axiomatic performance of the considered axioms is not an indicator of retrieval effectiveness. One reason for this may be that the shorter queries in the MSMarco dataset (5.99 on average) may not be long enough to avoid inherent ambiguity of language (polysemous words, synonyms and so on) [23], making the $\overline{\texttt{TFC}}$ and $\overline{\texttt{M-TDC}}$ axioms less important in the sense that their heuristics are less important compared to their importance for properly finding a relevant document in WikiPassageQA. This possible explanation is supported by the RM3 model, which employs query expansion using pseudo-relevance feedback to tackle such problems, outperforming the other baselines, albeit by a short margin. Hence, it would be interesting to see, how axiomatic performances on diagnostic datasets for semantic axioms compare to the retrieval effectiveness of models that we obtained for MSMarco. However, we leave this here as future work.

When we look at the relation between retrieval effectiveness and axiomatic performance per axiom, we obtain that none of the axioms seems to be a proper diagnostic for retrieval effectiveness as displayed in Fig. 4.3: for each of the plots there seems to be either a negative trend ($\overline{\texttt{TFC1}}$, $\overline{\texttt{M-TDC}}$) or no trend at all ($\overline{\texttt{TFC2}}$, $\overline{\texttt{LNC2}}^{Test}$). However, these findings do match our expectations discussed in Section 3.5.3: the $\overline{\texttt{TFC1}}$ and $\overline{\texttt{M-TDC}}$ axioms encapsulate heuristics that do not seem to be important in MSMarco.



(a) $\overline{\texttt{TFC1}}$ (b) $\overline{\texttt{TFC2}}$ (c) $\overline{\texttt{M-TDC}}$

(d) $\overline{\texttt{LNC2}}^{TEST}$

Figure 4.3: Axiomatic performance (fraction of fulfilled diagnostic instances) versus retrieval effectiveness (MAP) of traditional (♦) and neural (•) IR models per axiom for the `MSMarco` dataset. Figures also include a (dashed gray) trendline[18].

[18] Obtained with `numpy.polyfit`.

4.3. The Impact of Document Length Differences

The original axioms TFC1, TFC2 and TDC (but not LNC2) require strict document length equality ($|\boldsymbol{d_i}| - |\boldsymbol{d_j}| = 0$). In our axiom conversion (detailed in Section 3.3) we adopted a parameter in $\overline{\texttt{TFC1}}$, $\overline{\texttt{TFC2}}$ and $\overline{\texttt{TDC}}$ to regulate this difference ($|\boldsymbol{d_i}| - |\boldsymbol{d_j}| \leq \delta$). Instead of 0, this parameter was not constrained to a specific value (since this comes at the risk of obtaining less diagnostic instances). Hence the documents $\boldsymbol{d_i}, \boldsymbol{d_j}$ in our diagnostic instances may have significant differences in length. In this section we research the impact of such document length differences on axiomatic performance and answer presence.

4.3.1. Impact on axiomatic performance

In this section we elaborate upon the sensitivity of models (based upon their axiomatic performance) to length differences in diagnostic instances.

Methodology

Fig. 4.4 displays the fraction of diagnostic instances fulfilled by BM25 and QL for the $\overline{\texttt{TFC2}}$ axiom obtained from the WikiPassageQA dataset versus the maximum document length difference δ in documents $\boldsymbol{d_i}, \boldsymbol{d_j}, \boldsymbol{d_k}$ in the diagnostic instances on a logarithmic scale. This simple plot already reveals that the axiomatic performance of these traditional IR models can be (very) sensitive to document length differences: whereas both models fulfill all diagnostic instances when $\delta = 0$, QL fulfills close to 40% less diagnostic instances than BM25 if $\delta = 10,000$.

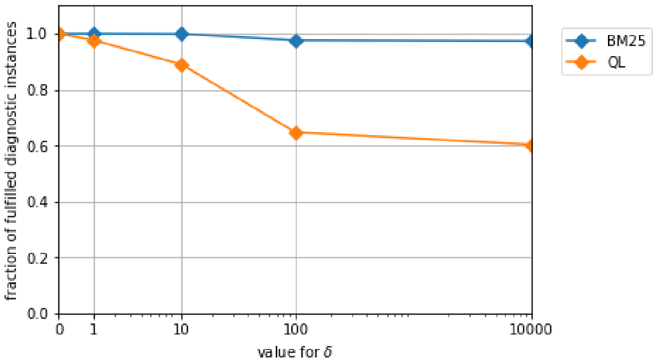


Figure 4.4: Fraction of fulfilled $\overline{\texttt{TDC}}$ instances obtained from WikiPassageQA per maximum document length difference (expressed in an absolute value for δ): document length differences can impact axiomatic performance.

In the next section, we will provide a closer look at the impact of axiomatic performance across our diagnostic datasets for different axioms. Since the WikiPassageQA and MSMarco dataset consist of documents that follow a different length distribution (recall Fig. 3.4), we do not choose δ to be an absolute value (e.g. 0, 1, 10, 100, 1000 word(s)) as was done in Fig. 4.4. Instead, we adopt a relative parameter δ^*, in similar spirit as Hagen et al. [53], as expressed in Eq. 4.1. Note that the parameter δ^* we adopt here is equal to the δ parameter introduced in our axioms multiplied by the maximum document length in a diagnostic instance: $\delta^* = \delta * \max\{|\boldsymbol{d_i}|, |\boldsymbol{d_j}|\}$. Moreover, for $\overline{\texttt{TFC2}}$ we also incorporate $|\boldsymbol{d_k}|$ by adopting $|\boldsymbol{d_i}| = \min\{|\boldsymbol{d_i}|, |\boldsymbol{d_j}|, |\boldsymbol{d_k}|\}$ and $|\boldsymbol{d_j}| = \max\{|\boldsymbol{d_i}|, |\boldsymbol{d_j}|, |\boldsymbol{d_k}|\}$.

$$\frac{abs(|\boldsymbol{d_i}| - |\boldsymbol{d_j}|)}{\max\{|\boldsymbol{d_i}|, |\boldsymbol{d_j}|\}} \leq \delta^* \tag{4.1}$$

We will adopt two sets of values for δ^*. We will consider a *low range*: $0 - 0.1$ with steps of 0.01 (i.e. no document length difference up to a difference of 10% of the largest document with steps of 1%), and a *high range*: $0.1 - 1.0$ with steps of 0.1 (i.e. 10% up to a difference of 100% of the largest document with steps of 10%). By doing so, we can obtain what happens if we stay closest to the axioms without document relaxation ($\delta^* = 0$), as well as what happens in the long run if we allow one document to be twice as long as the other document ($\delta^* = 1.0$). In fact, all relative differences in document length in our diagnostic instances across both datasets are below 1.0, hence at $\delta^* = 1.0$ we will find the axiomatic performances as displayed in Table 4.5 and 4.6 respectively.

Additionally, we note that we should keep track of the amount of diagnostic instances per value for δ^*, since we need diagnostic instances to be able to diagnose anything at all.

Results

Figures 4.5 and 4.6 respectively display the fraction of fulfilled instances per value for δ^* for the `WikiPassageQA` and `MSMarco` datasets per axiom (excluding `LNC2` which does not require a strict document length equality). These figures also display the amount of diagnostic instances for the displayed values for δ^* (at the top of each plot). From these figures we observe three patterns:

1. *The axiomatic performance of **traditional models** seems to **decrease** when document length differences increase*;
   We can clearly observe this pattern for the $\overline{\texttt{TFC1}}$ instances on the `MSMarco` dataset (Fig. 4.6a). To a lesser extent, we can also observe this pattern on the $\overline{\texttt{TFC1}}$ and $\overline{\texttt{TFC2}}$ instances from `WikiPassageQA` (resp. Fig. 4.5a and 4.5b, excluding BM25) as well as the $\overline{\texttt{M-TDC}}$ instances from `MSMarco` (Fig. 4.6c, excluding RM3).

2. *The axiomatic performance of **neural models** seems to be **less impacted** by document length differences*;
   We can find some small deviations in axiomatic performance of some neural models: for example, the axiomatic performance of DRMM and MV-LSTM on $\overline{\texttt{TFC1}}$ and $\overline{\texttt{TFC2}}$ in `WikiPassageQA` (resp. Fig 4.5a and 4.5b) increases when δ^* increases, but their axiomatic performance on $\overline{\texttt{M-TDC}}$ in `WikiPassageQA` and on $\overline{\texttt{TFC1}}$ in `MSMarco` (resp. Fig 4.5c and 4.6a) decreases when δ^* increases. However, overall, the performance of the neural models seems to be less impacted by document length difference, compared to the impact it has on the axiomatic performance of traditional models.

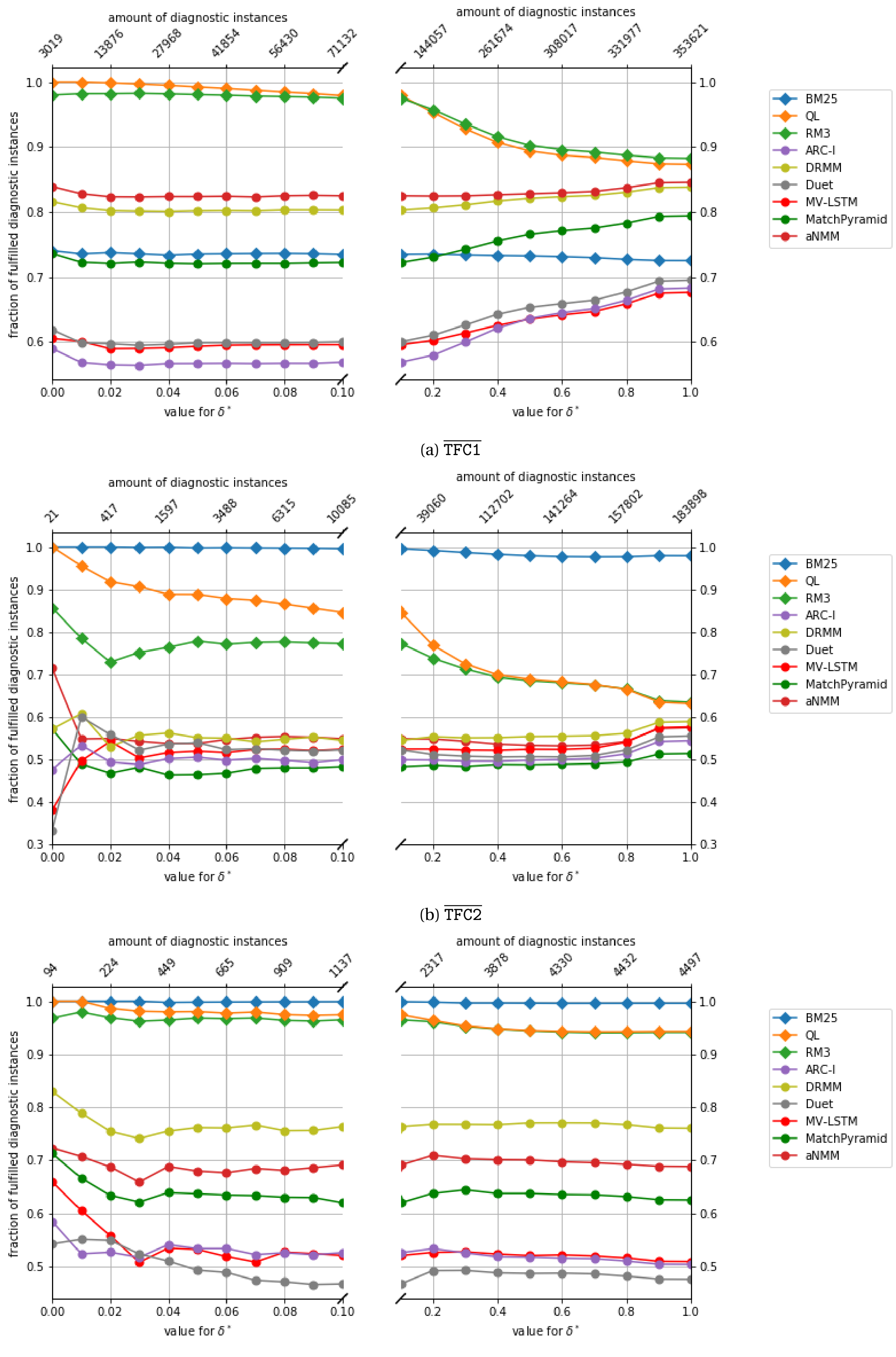
3. *The axiomatic performance of **all models** seems to **converge** when document length differences (δ^*) increase*;
   In other words, the difference between the axiomatic performance of most models decreases over the coarse of increasing $\delta^* = 0$ to $\delta^* = 1.0$. We can clearly see this pattern for the $\overline{\texttt{TFC2}}$ axiom instances obtained from `MSMarco` (Fig. 4.6b), but also in the $\overline{\texttt{TFC1}}$ instances in both datasets (Fig. 4.5a and 4.6a, resp.) and the $\overline{\texttt{M-TDC}}$ instances in `MSMarco` (Fig. 4.5c). However this pattern is not obtained for the $\overline{\texttt{M-TDC}}$ instances in the `WikiPassageQA` dataset (Fig. 4.5c);

We note that the third observation also relates to the first two observations: when δ^* increases, the traditional models generally achieve lower axiomatic performance while the neural models generally are not impacted much. Hence, since the traditional models typically achieved higher axiomatic performance–which now decreases and therefore comes closer to the axiomatic performance of neural models—the axiomatic performances of all models show a converging pattern.

From these plots it is however difficult to obtain what a reasonable value for δ^* — i.e. a value that allows to obtain a proper amount instances while staying properly close to $\delta^* = 0$—would be. Whereas putting δ^* to 0 can result in having no diagnostic instances (e.g. for $\overline{\texttt{TFC2}}$ instances from `MSMarco`), putting δ^* to 0.2 swaps the order of the axiomatic performance of some models (compared to $\delta^* = 0$) across all considered axioms.

We therefore provide a closer look at the *deviations* from the "golden" $\delta^* = 0$ values. Figures 4.7 and 4.8 respectively display the *absolute difference* in the fraction of fulfilled instances per value for δ^* for the `WikiPassageQA` and `MSMarco` datasets per axiom. We again exclude `LNC2` which does not require a strict document length equality, but now also exclude $\overline{\texttt{TFC2}}$ for `MSMarco` since we have not obtained diagnostic instances for $\overline{\texttt{TFC2}}$ from `MSMarco` for which $\delta^* \leq 0.01$ (hence we can not obtain the deviation from $\delta^* = 0$). From these figures we can however also not directly obtain a proper value for δ^*[19]. Neither do these images provide reason to consider the $\delta^* = 0.10$ adopted in [52] to be improper. Nevertheless, we can from these images observe that *the **larger** the **document length difference δ^***, the **larger** the **deviation in axiomatic performance** compared to the golden values obtained at $\delta^* = 0$.*

[19] For completeness we note that it may seem like $\delta^* = 0.10$ is a proper value as the absolute differences in axiomatic performance in the right hand of the plots are larger than those of the left hand. However, whereas the left hand of the plots considers steps of 0.01, the right hand of the plots considers steps of 0.10. Hence, knowing that the deviations in the right hand of the plot are expected to be an order of magnitude larger, we do not observe $\delta^* = 0.10$ to specifically be a suitable value for δ^*.

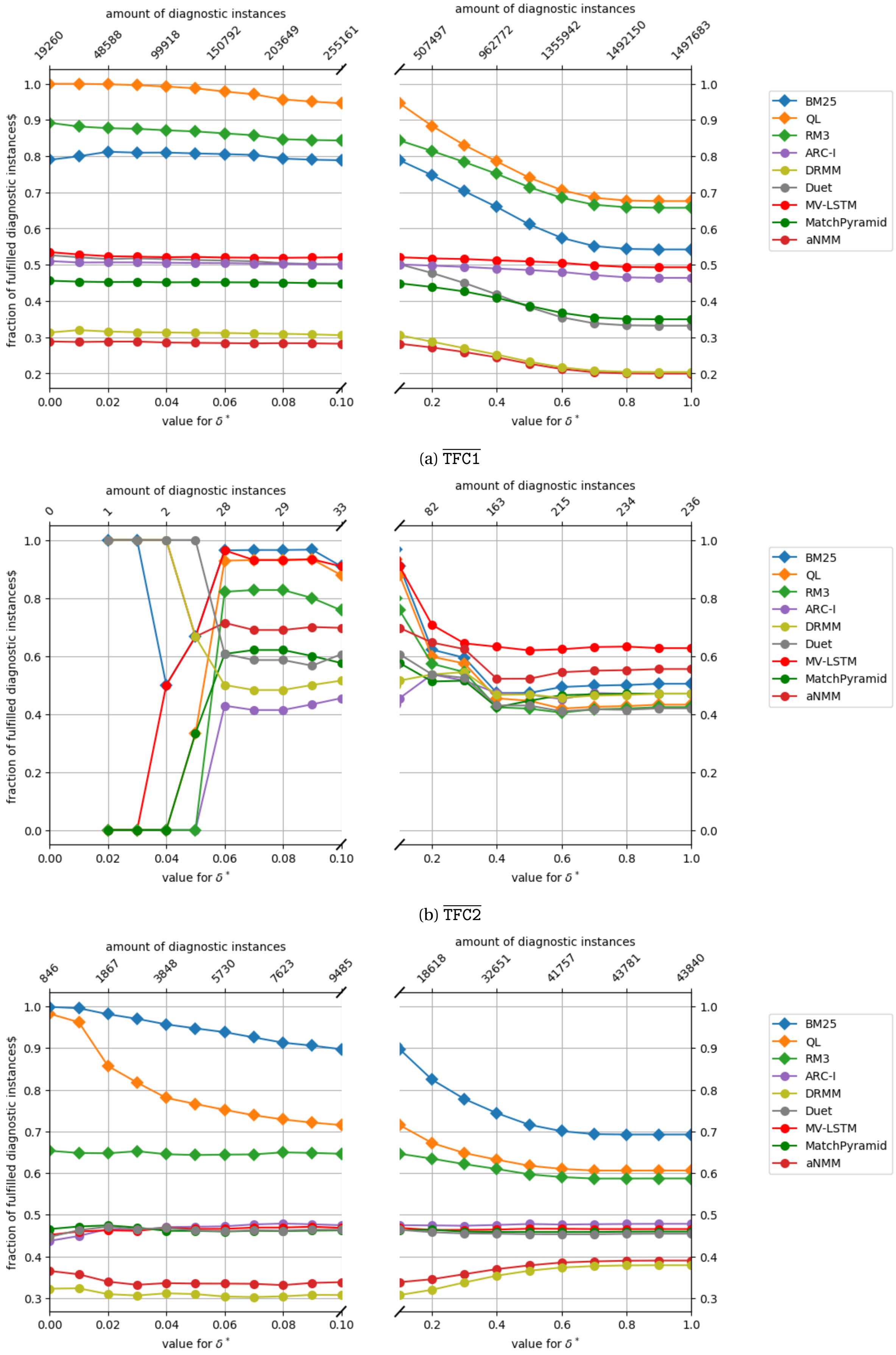


(a) $\overline{\texttt{TFC1}}$

(b) $\overline{\texttt{TFC2}}$

(c) $\overline{\texttt{M-TDC}}$

Figure 4.5: Impact of document length differences on axiomatic performance of traditional (♦) and neural (•) IR models per axiom: measured in the fraction of fulfilled diagnostic instances (vertical axis) per value of δ^* (horizontal axis) for the `WikiPassageQA` dataset. Figures also display the amount of diagnostic instances per value of δ^* (horizontally in the top of the figures).

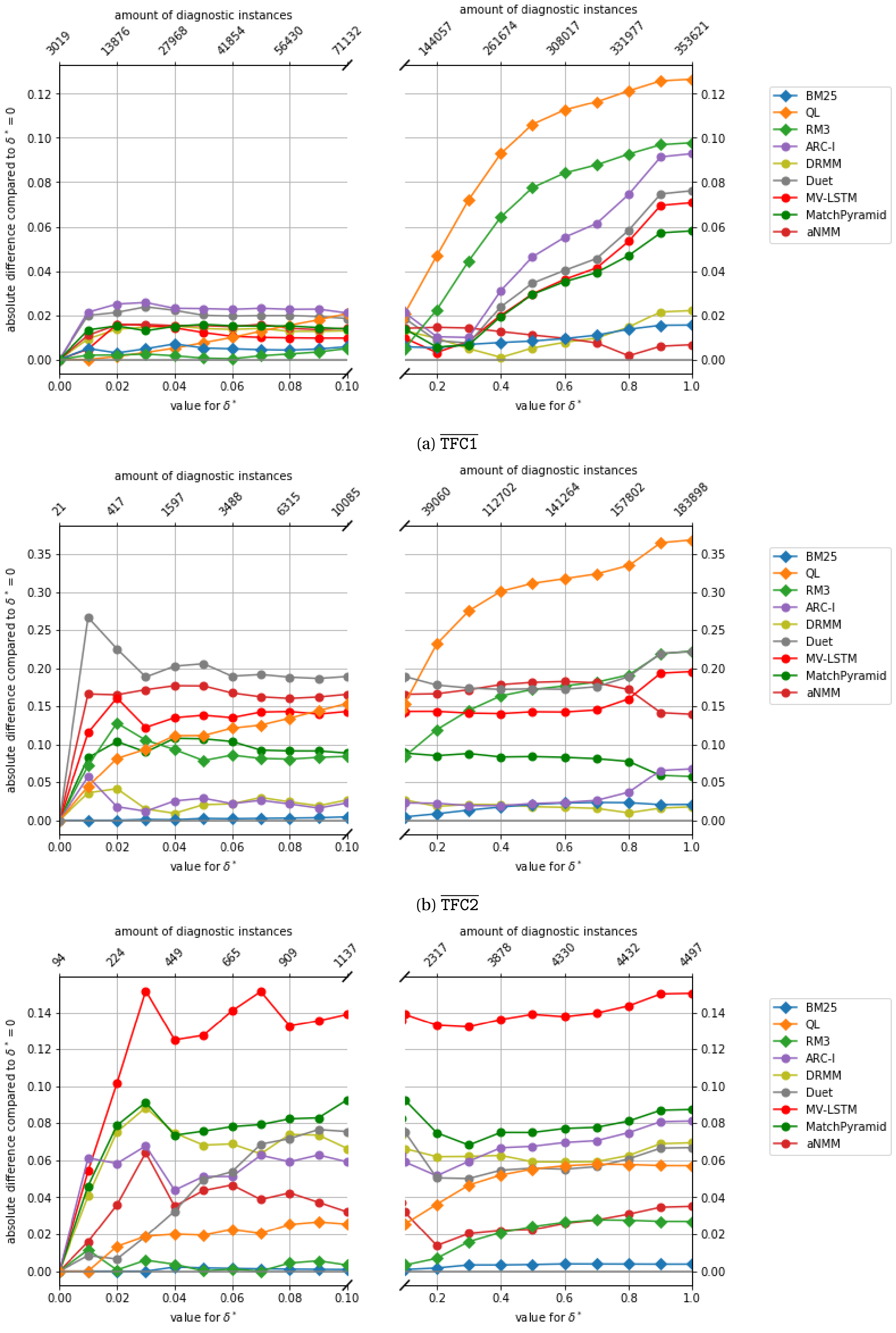


(a) $\overline{\texttt{TFC1}}$

(b) $\overline{\texttt{TFC2}}$

(c) $\overline{\texttt{M-TDC}}$

Figure 4.6: Impact of document length differences on axiomatic performance of traditional (♦) and neural (•) IR models per axiom: measured in the fraction of fulfilled diagnostic instances (vertical axis) per value of δ^* (horizontal axis) for the `MSMarco` dataset. Figures also display the amount of diagnostic instances per value of δ^* (horizontally in the top of the figures).

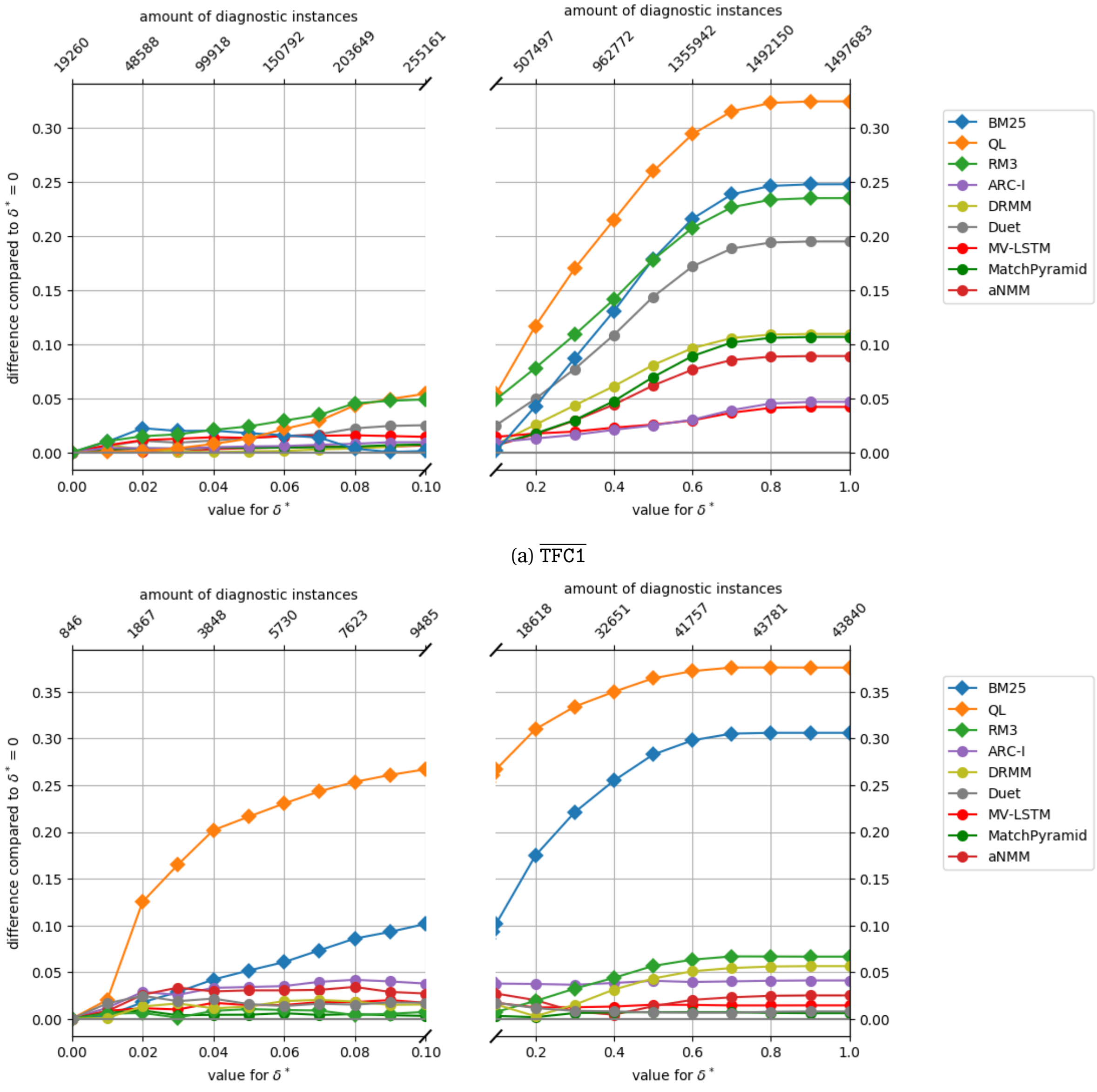


(a) $\overline{\texttt{TFC1}}$

(b) $\overline{\texttt{TFC2}}$

(c) $\overline{\texttt{M-TDC}}$

Figure 4.7: Impact of document length differences on axiomatic performance of traditional (♦) and neural (•) IR models per axiom: measured in the *absolute difference in fraction of fulfilled diagnostic instances compared to* $\delta^* = 0$ (vertical axis) per value of δ^* (horizontal axis) for the `WikiPassageQA` dataset. Figures also display the amount of diagnostic instances per value of δ^* (horizontally in the top of the figures).



(a) $\overline{\texttt{TFC1}}$

(b) $\overline{\texttt{M-TDC}}$

Figure 4.8: Impact of document length differences on axiomatic performance of traditional (♦) and neural (•) IR models per axiom (excluding $\overline{\texttt{TFC2}}$): measured in the *absolute difference in fraction of fulfilled diagnostic instances compared to* $\delta^* = 0$ (vertical axis) per value of δ^* (horizontal axis) for the `MSMarco` dataset. Figures also display the amount of diagnostic instances per value of δ^* (horizontally in the top of the figures)

Impact on answer presence

We further research the impact of document length difference on the presence of relevant documents among the documents $\boldsymbol{d_i}$, $\boldsymbol{d_j}$ (, $\boldsymbol{d_k}$) in the diagnostic instances. Tables 4.7 and 4.8 respectively present the presence of the various combinations of documents per value for $\delta^* = [0, 0.01, 0.10, 1.0]$ for diagnostic instances obtained from the test splits of the `WikiPassageQA` and `MSMarco` datasets.

Overall, we view that the relation between the "relevant ≥ non-relevant" pairs and the "non-relevant ≥ relevant" pairs stays the same over the various values for δ^*, across axioms and datasets. The only large deviation is obtained for the $\overline{\texttt{TFC1}}$ instances in the `MSMarco` dataset (see the first few rows of Table 4.8): whereas both type of pairs are equally found among the lower δ^* values (0, 0.01, 0.1), a large deviation is found in the step from $\delta^* = 0.1$ (5,050 pairs versus 5,391 pairs) to $\delta^* = 1.0$ (28,968 pairs versus 52,472 pairs). Hence, this may give reason to put δ^* lower than 0.1.

	$\boldsymbol{d_i \succ d_j}$	$\boldsymbol{\delta^* = 0}$		$\boldsymbol{\delta^* = 0.01}$		$\boldsymbol{\delta^* = 0.10}$		$\boldsymbol{\delta^* = 1.0}$	
$\overline{\texttt{TFC1}}$	**relevant ≻ relevant**	0	0.0%	3	0.0%	22	0.0%	96	0.0%
	relevant ≻ non-relevant	80	2.6%	230	3.0%	2,322	3.3%	10,937	3.1%
	non-relevant ≻ relevant	20	0.7%	53	0.7%	412	0.6%	1,809	0.5%
	non-relevant ≻ non-relevant	2,919	96.7%	7,463	96.3%	68,376	96.1%	340,779	96.4%
	total	3,019	100.0%	7,749	100.0%	71,132	100.0%	353,621	100.0%
$\overline{\texttt{TFC2}}$									
	$S(\boldsymbol{q},\boldsymbol{d_j}) - S(\boldsymbol{q},\boldsymbol{d_i}) > S(\boldsymbol{q},\boldsymbol{d_k}) - S(\boldsymbol{q},\boldsymbol{d_j})$	$\boldsymbol{\delta^* = 0}$		$\boldsymbol{\delta^* = 0.01}$		$\boldsymbol{\delta^* = 0.10}$		$\boldsymbol{\delta^* = 1.0}$	
	total	21	100.0%	135	100.0%	10,085	100.0%	183,898	100.0%
$\overline{\texttt{M-TDC}}$	$\boldsymbol{d_i \succeq d_j}$	$\boldsymbol{\delta^* = 0}$		$\boldsymbol{\delta^* = 0.01}$		$\boldsymbol{\delta^* = 0.10}$		$\boldsymbol{\delta^* = 1.0}$	
	relevant ⪰ relevant	0	0.0%	0	0.0%	0	0.0%	0	0.0%
	relevant ⪰ non-relevant	0	0.0%	3	2.0%	17	1.5%	50	1.1%
	non-relevant ⪰ relevant	1	1.1%	2	1.4%	11	1.0%	56	1.2%
	non-relevant ⪰ non-relevant	93	98.9%	142	96.6%	1,109	97.5%	4,391	97.6%
	total	94	100.0%	147	100.0%	1,137	100.0%	4,497	100.0%

Table 4.7: Presence of relationships as prescribed per axiom for the diagnostic datasets obtained from the **test** split of the `WikiPassageQA` dataset.

	$\boldsymbol{d_i \succ d_j}$	$\boldsymbol{\delta^* = 0}$		$\boldsymbol{\delta^* = 0.01}$		$\boldsymbol{\delta^* = 0.10}$		$\boldsymbol{\delta^* = 1.0}$	
$\overline{\texttt{TFC1}}$	**relevant ≻ relevant**	2	0.0%	2	0.0%	32	0.0%	160	0.0%
	relevant ≻ non-relevant	485	2.5%	513	2.2%	5,050	2.0%	28,968	1.9%
	non-relevant ≻ relevant	466	2.4%	510	2.2%	5,391	2.1%	52,472	3.5%
	non-relevant ≻ non-relevant	18,307	95.1%	22,324	95.6%	244,688	95.9%	1,416,083	94.6%
	total	19,260	100.0%	23,349	100.0%	255,161	100.0%	1,497,683	100.0%
$\overline{\texttt{TFC2}}$									
	$S(\boldsymbol{q},\boldsymbol{d_j}) - S(\boldsymbol{q},\boldsymbol{d_i}) > S(\boldsymbol{q},\boldsymbol{d_k}) - S(\boldsymbol{q},\boldsymbol{d_j})$	$\boldsymbol{\delta^* = 0}$		$\boldsymbol{\delta^* = 0.01}$		$\boldsymbol{\delta^* = 0.10}$		$\boldsymbol{\delta^* = 1.0}$	
	total	0	0.0%	0	0.0%	33	100.0%	236	100.0%
$\overline{\texttt{M-TDC}}$	$\boldsymbol{d_i \succeq d_j}$	$\boldsymbol{\delta^* = 0}$		$\boldsymbol{\delta^* = 0.01}$		$\boldsymbol{\delta^* = 0.10}$		$\boldsymbol{\delta^* = 1.0}$	
	relevant ⪰ relevant	1	0.1%	1	0.1%	1	0.0%	2	0.0%
	relevant ⪰ non-relevant	15	1.8%	16	1.7%	127	1.3%	611	1.4%
	non-relevant ⪰ relevant	28	3.3%	28	3.0%	254	2.7%	1,116	2.5%
	non-relevant ⪰ non-relevant	802	94.8%	879	95.1%	9,103	96.0%	42,111	96.1%
	total	846	100.0%	924	100.0%	9,485	100.0%	43,840	100.0%

Table 4.8: Presence of relationships as prescribed per axiom for the diagnostic datasets obtained from the **test** split of the `MSMarco` dataset.

4.4. Fixing Neural IR Models

Given a diagnosis of a deep model, one can try to "fix" it, analogous to how traditional IR models have been fixed based upon their axiomatic performance by adapting their retrieval function (as was discussed in Section 2.5). When considering neural approaches, we can however no longer adapt the retrieval function as this retrieval function is now spread over the deep architecture. Nevertheless, we *can* adapt parts of the process that eventually make up the deep model. We here specifically propose to augment the training data of a deep net with previously created diagnostic instances. Hence, we propose a novel methodology for improving the axiomatic performance and retrieval effectiveness of neural models and briefly research it in this section. In the remainder of this section, we first introduce a background of our approach (Section 4.4.1), followed by an introduction of our methodology (Section 4.4.2) and a brief experiment we have conducted (Section 4.4.3)

4.4.1. Background

Despite the proposal of numerous novel neural IR models, little attention has been given to addressing the shortcomings of existing neural models (other than by proposing a novel model). The lack of approaches to improve a neural model may very well stem from the shortage of means to analyze neural IR models (as discussed in Section 2.4.1): researchers may have been inclined to propose complete new architectures, rather than tackling (largely unknown) issues of existing models.

To address issues of individual retrieval models we here propose a novel method based on training data augmentation. As the name indicates, training data augmentation considers enriching a regular training data set with additional instances, with the aim of improving the models performance. Our employment of training data augmentation in essence combines ideas from two branches of related works, respectively on weak supervision in neural IR and axiomatic re-ranking. In one of the few works that has explored weak supervision in neural IR, Dehghani et al. [35] showed that neural models that are trained on weak supervision signals can achieve impressive performance improvements over a weak supervisor (BM25). Moreover, they also showed that pre-training a neural IR model on large amounts of weak supervision signals (and a small amount of supervised signals), can achieve performance improvements over training under only weak supervision or full supervision. Note that in the latter experiment the training data is thus augmented with instances that are labeled by a weak supervisor. In a different study (on axiomatic re-ranking), Hagen et al. [53], showed that the retrieval effectiveness of traditional IR models could be improved through re-ranking their ranked output in an axiomatic manner post retrieval (as discussed in Section 2.5). Combining both works in our approach, we do not use axiomatic re-ranking as a separate final component. Instead, we propose to utilize axioms as a tool to augment the input of a model: we employ axioms as weak labelers to obtain instances for training data augmentation.

Recently, Rosset et al. [108] have considered the direct incorporation of axioms in the loss function as discussed in Section 2.5.2. Whereas their method encompasses a general approach to augment the training scheme of any neural model, we here focus on addressing the specific individual shortcomings of neural models through including diagnostic instances (we already obtained) in the training set while maintaining the same loss function.

Training data augmentation is typically employed to obtain more training data (e.g. when there is little training data available) [35, 108]. A model is then trained on both the original and the newly obtained data. However, due to issues in implementing this approach in MatchZoo[20], we do not follow this approach. Instead, we replace random instances in the regular training data of the model with weakly labeled instances, so that the total amount of instances on which a model is trained stays the same.

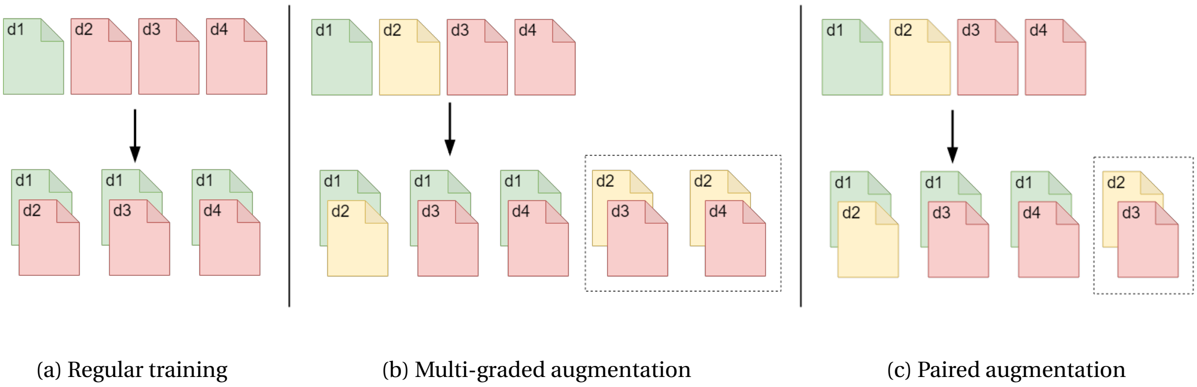
We nevertheless hypothesize that the adoption of augmented instances in the training set can improve a model's performance on the axiom used as a weak labeler. We furthermore hypothesize that this also improves the retrieval effectiveness of the model.

[20] Although we can not refer to an issue, it is to the best of our knowledge not possible to resume training in MatchZoo (e.g. to first train a model on regular data and then continue training the model on data labeled by a weak supervisor or the other way around), neither is it trivial to program which training data is used at which iteration.

4.4.2. Methodology

We can use an axiom as a weak labeler that assigns labels to dataset instances based upon the heuristic the axiom encapsulates. Recall that an axiom typically ($\overline{\texttt{TFC2}}$ is an exception here) prescribes that a document $\boldsymbol{d_i}$ should receive a higher score than a document $\boldsymbol{d_j}$. We can extract weak labels from such relationships: e.g., we can give $\boldsymbol{d_i}$ label 1 and give $\boldsymbol{d_j}$ label 0 - we could then label relevant passages with e.g. a label 2. Intuitively it makes sense to adopt these multi-graded relevance labels: a **strong label** 2 indicates that a document is more relevant than *any* other *non-relevant* document $\boldsymbol{d_j}$, whereas a **weak label** 1 only represents a relationship of a document and *one or more* (*relevant or non-relevant*) documents.

The adoption of such labels, allows us to augment the regular training data. Similar as before, we train a model with a query and a pair of documents following the pair-wise training regime. Now, let us consider a simple example in which we have one query and four documents that are to be ranked for this query. Under the regular training scheme, we would train on a query and pairs of a relevant and non-relevant document, which are graphically displayed left in Fig. 4.9. We now augments these pairs with pairs in which one document should be scored above (i.e. is deemed more relevant) another document based upon a certain axiom. Let us assume that $\boldsymbol{d_2}$ should be scored higher than $\boldsymbol{d_3}$ according to some axiom. In the simplest case, strictly following the adopted multi-graded labels, we would train on a query and all regular pairs of a relevant and non-relevant documents (note that these instances can have labels 2 and 1 as well as 2 and 0) and augmented pairs of documents (e.g. with labels 1 and 0), as displayed in the middle in Fig. 4.9. This strategy is similar to how BM25 was used as a weak supervisor by Dehghani et al. [35]. However, different from BM25, our weak supervisor provides us with pairwise labels, rather than listwise labels or scores. Hence, in the example displayed in this figure, we actually do not know if we would want a model to score $\boldsymbol{d_2}$ above $\boldsymbol{d_4}$: we only know that some axiom prescribed that $\boldsymbol{d_2}$ should be scored higher than $\boldsymbol{d_3}$. Therefore, it would make more sense to only include $\boldsymbol{d_2}$, $\boldsymbol{d_3}$, as displayed right in Fig. 4.9.



(a) Regular training (b) Multi-graded augmentation (c) Paired augmentation

Figure 4.9: Regular training instances (4.10a) can be augmented (as displayed in a dotted box) by including instances obtained with a weak supervisor (4.10b), which, when obtained from axioms, intuitively should be paired (4.10c). Relevant documents are displayed in green, non-relevant documents in red and document $\boldsymbol{d_2}$ that should be ranked above document $\boldsymbol{d_3}$ according to some axiom is displayed in yellow. For simplicity we have omitted queries in the training instances.

However, we note that we have to avoid two issues that are not visible in the carefully chosen example we have discussed. Specifically, following our methodology, we can obtain two types of instances that we should filter out. Firstly, we exclude instances where a non-relevant document $\boldsymbol{d_i}$ (labeled 1) should be scored higher than a relevant document $\boldsymbol{d_j}$ (labeled 2) according to some axiom, as this is counter-intuitive to the strong relevance label. We also exclude instances in which a relevant document $\boldsymbol{d_i}$ (labeled 2) should be scored higher than another relevant document $\boldsymbol{d_j}$ (also labeled 2), as this is counter-intuitive to the multi-graded labels: both documents have a label 2 and are therefore considered equally relevant[21].

Distribution

Next to the creation of instances to augment the training data, we have to decide upon the distribution in the training data. More specifically said, we need to decide upon the distribution of original training pairs (of a relevant and non-relevant document) and our novel axiomatic training pairs (of one document that should

[21] One could argue that for our purpose of fixing a model, we actually could use such instances for training. However, this may actually drift a model away from recognizing $\boldsymbol{d_j}$ as relevant, since such instances inversely also tell that a relevant document $\boldsymbol{d_j}$ should receive a lower score than another document.

be scored higher than the other document according to a specific axiom). We namely have to keep in mind that a model can become overfit when trained on augmented data. For example, if we only train a model on triplets $\boldsymbol{q}$, $\boldsymbol{d_i}$, $\boldsymbol{d_j}$ with $\overline{\texttt{TFC1}}$ prescribing that $S(\boldsymbol{d_i}, \boldsymbol{q}) > S(\boldsymbol{d_j}, \boldsymbol{q})$, the trained model may very well become overfit with regard to the weak supervisor [35] and learn to simply mimic a TF formula. However, it is difficult to determine what a suitable distribution of such instances would be. Dehghani et al. [35] employ weak supervision in all instances in either the full training or the pre-training set (which considers the predominant part of the complete training) of their models (as was also done in [139] for neural approaches on the query performance prediction task and [20] for improving LTR approaches)[22]. However, since our weak supervisor is a single axiom rather than a whole model, we expect that training on only weak supervision signals will not improve the performance of a model over the performance obtained after regular training, we here adopt a 1:1 ratio as a starting point (but also experiment with a unconstrained ratio).

Development and test sets

Since we are ultimately still interested in the effectiveness of the model to retrieve an answer, we revert to the two label (2 = relevant, 0 = non-relevant) setting in the dev and test set, as we only have answers and non-answers in the “real-world” setting.

4.4.3. Experiment: improving Duet on $\overline{\texttt{TFC1}}$

Given the (relative) poor performance of Duet on the $\overline{\texttt{TFC1}}$ axiom in `WikiPassageQA` (it fulfilled 69% of the instances) and the fact that this axiom relatively often leads to an answer (in a 7:1 ratio, as discussed in Section 3.5.3), we train this model on a training set that is augmented with instances obtained with the $\overline{\texttt{TFC1}}$ axiom as a weak labeler.

The default training dataset with the regular relevance labels consist of 156M instances, among which we can only find 82K pairs of a relevant and non-relevant document in which the $\overline{\texttt{TFC1}}$ axiom holds. Through adopting the proposed multi-graded labels, we can obtain 2.6M pairs of non-relevant documents that fulfill the conditions of $\overline{\texttt{TFC1}}$. Hence, we can sample from this vast amount of instances to augment the original training set.

In the following paragraphs we discuss which of these pairs were included. The results of the various experiments are displayed in Table 4.9[23]. An overview with more detailed numbers can be found in Appendix B.2. Per experiment, we discuss the adopted training scheme, the results and (possible) explanation in subsequent paragraphs.

	Retrieval effectiveness			**Performance per axiom**		
Training data	MAP	MRR	P@5	$\overline{\texttt{TFC1}}$	$\overline{\texttt{TFC2}}$	$\overline{\texttt{M-TDC}}$
^{1} answers	0.25^{2}	0.29^{2}	$\mathbf{0.10}^{2}$	0.69	$\mathbf{0.56}^{2,3,4}$	0.48
^{2} answers + $\overline{\texttt{TFC1}}_{multi\text{-}graded}$	0.20	0.22	0.07	$0.75^{1,4}$	0.50^{4}	0.49
^{3} answers + $\overline{\texttt{TFC1}}_{paired,1:1}$	$\mathbf{0.27}^{2}$	$\mathbf{0.31}^{2}$	$\mathbf{0.10}^{2}$	$\mathbf{0.81}^{1,2,4}$	$0.54^{2,4}$	$\mathbf{0.51}^{1,2}$
^{4} answers + $\overline{\texttt{TFC1}}_{paired,1:1,\delta^*\leq 0.1}$	0.26^{2}	$\mathbf{0.31}^{2}$	$\mathbf{0.10}^{2}$	0.70^{1}	0.49	$\mathbf{0.51}^{1}$

Table 4.9: Overview of retrieval effectiveness and fraction of fulfilled axiom instances for the Duet model trained on the original and augmented versions of the `WikiPassageQA` dataset. For measuring statistical significance, we employed the Wilcoxon test and McNemar test with $p < 0.05$ on respectively measures for retrieval effectiveness and axiomatic performance.

[22] It is unclear which ratio is employed in [108].

[23] Note that we have not included the $\overline{\texttt{LNC2}}$ axiom in our diagnosis per axiom. Next to the fact that we obtained that a model’s adherence to this axiom has not shown to be a good indicator for retrieval effectiveness for the `WikiPassageQA` dataset (as discussed in Section 4.2.1), inclusion of the axiom would require separate runs that include artificial data, while our focus here is on improving the performance of a model with respect to another axiom and regarding retrieval effectiveness.

Multi-graded augmentation

As a first experiment, we simply follow the multi-graded augmentation displayed in the middle of Figure 4.9: we consider relevant documents labeled 2 to be more relevant than documents labeled 1 which we consider more relevant than documents labeled 0. The results of this experiment are displayed in the second row of Table 4.9. We see that the performance for the $\overline{\texttt{TFC1}}$ axiom indeed increases, but the training data augmentation has actually been detrimental to the model's retrieval effectiveness.

This problem may stem from two issues. First, there are many more documents with label 1 than documents with either label 2 or 0. Hence, a model gets trained on many more pairs $(1,0)$ than $(2,0)$, while there are also not that many instances $(2,1)$ (see Table 3.10). In addition, the model may not learn much from a pair $(\boldsymbol{d_i},\ \boldsymbol{d_j})$ labeled $(1,0)$ in case $\boldsymbol{d_i}$ was assigned a label 1 since $\overline{\texttt{TFC1}}$ states it should be ranked higher than another document $\boldsymbol{d_k}$ with $\boldsymbol{d_j} \neq \boldsymbol{d_k}$. Hence, we investigate approaches beyond this facile approach.

Paired augmentation with equal distributions

Now let us consider a more reasonable approach, in which we augment the training data only with document pairs $(\boldsymbol{d_i},\ \boldsymbol{d_j})$ labeled $(1,0)$ in case $\overline{\texttt{TFC1}}$ prescribes $\boldsymbol{d_i}$ should retrieve a higher score than $\boldsymbol{d_j}$. Additionally, we make sure that we train on pairs $(1,0)$ in an equal amount as $(2,1)$ and $(2,0)$. Note that in the non-augmented training scheme we would train on $(2,1)$ and $(2,0)$ instances. Hence, we now train on the default instances and axiomatic instances in a 1:1 ratio.

The results of this experiment are displayed in the third row of Table 4.9. We see that the performance for the $\overline{\texttt{TFC1}}$ axiom now increases even more than in the previous experiment, while not lowering performance on the other axioms. Moreover the Duet model now achieves even higher performance than e.g. BM25 on the diagnostic instances for this axiom (0.81 versus 0.73). Looking at the retrieval effectiveness, we find that although the training data augmentation has been beneficial to the model's retrieval effectiveness, with an increase in both MAP and MRR (0.02 in both), but not in an significant manner.

Paired augmentation with equal distributions and constrained length differences

Now let us further develop our approach by also accounting for the difference in document length within instances by filtering out any diagnostic instance for which the document length difference expressed in δ^* (introduced in Section 4.3) is larger than 0.1. We choose this value as it provides us with instances that are closer to the original axiom definition (in the sense that they deviate less from the original constraint that $|\boldsymbol{d_i}| = |\boldsymbol{d_j}|$), while we can still sample from a large enough set of instances to maintain a 1:1 ratio of regular and axiomatic instances in the training set.

We now find that the axiomatic performance has increased compared to the default setting, but only by 0.01, while lowering the performance on $\overline{\texttt{TFC2}}$ by 0.07 and increasing the performance on $\overline{\texttt{M-TDC}}$ by 0.03. Moreover, we find that the MAP and MRR respectively increase by 0.01 and 0.02, although again, not in a significant manner. One may think that the lower increase in $\overline{\texttt{TFC1}}$ can be explained by the fact that we are actually testing the model on instances for which we did not care about δ^*—in fact, for roughly 80% of the instances $\delta^* > 0.1$—but only train a model on instances for which $\delta^* \leq 0.1$. Looking at the axiomatic performance on diagnostic instances for $\overline{\texttt{TFC1}}$ so that $\delta^* \leq 0.1$, we however obtain similar numbers as presented in Table 4.9: 0.60 (regular training), 0.72 (multi-graded augmentation), 0.79 (paired augmentation), 0.71 (paired augmentation with constrained δ^*) - all significantly different. However, the strategy the includes a constrained δ^* is the only strategy under which the performance for these instances increased compared to the original instances considered for diagnosis, although marginally.

Concluding, we note that more research is needed into ways to address the issues of retrieval models obtained via an axiomatic diagnosis. Nevertheless, the proposed methodology in this chapter may provide a basis and the discussed experiments some preliminary results for such future work.

5 Conclusions and Future Work

In this chapter we conclude our work and elaborate upon various directions for future work, respectively in Section 5.1 and Section 5.2.

5.1. Conclusion

Over the past decade, the application of deep learning techniques has come to the forefront of Information Retrieval. Despite the large research efforts that have focused on proposing novel neural models, a number of issues have hindered the progress of neural IR. Among those issues, we have specifically addressed the issue of a lack of approaches to interpret and analyze neural IR models and posed the following research question: *How can we diagnose the strengths and weaknesses of neural IR approaches using axiomatic thinking?*

Knowing that the traditional axiomatic approaches are no longer viable to study neural IR models, we have taken inspirations from the NLP and Computer Vision communities to create so-called diagnostic datasets. These datasets can be used to determine what kind of search heuristics (encapsulated in axioms) neural models are able to learn. The creation of such diagnostic datasets, however, demanded the extension and subsequent relaxation of existing axioms to match instances we can find in realistic datasets. Since the creation of diagnostic datasets does not require a labeled dataset, we can apply the proposed pipeline to almost any dataset containing queries and documents. As the diagnostic datasets can be used to diagnose models based on their output, we can diagnose basically any IR model with them.

Using the proposed methodology, we have applied our diagnostic dataset creation pipeline to the `WikiPassageQA` and `MSMarco` corpora and evaluated three traditional baselines and six neural models on four established axioms for which we have proposed an extension and relaxation. From our experiments on `WikiPassageQA` we have learned that the proposed diagnostic approach can indeed identify strengths and weaknesses of neural models. From our experiments on `MSMarco` we have, on the other hand, learned that an axiomatic analysis based on the four axioms included in this work can not always capture factors that incur retrieval effectiveness. An interesting direction for future work is therefore to include other axioms encompassing heuristics on e.g. semantic matching which may be important for the `MSMarco` dataset.

Next to conducting diagnosis, we have researched the impact of allowing document length differences within diagnostic instances. We have studied how the axiomatic performance of neural models differs for various values of a parameter on the maximum allowed document length differences. As expected, we found that this deviation grows as the allowed difference in document length difference increases, although the performance of the studied neural models seem to generally be less impacted by document length differences than the studied traditional models.

Finally, we have also briefly researched how, given an axiomatic diagnosis, we can address weaknesses of neural models. We have proposed to augment training data with weakly supervised diagnostic instances that encapsulate a heuristic which a neural model has not learned under the regular training regime. As evaluated on the `WikiPassageQA` dataset by adding diagnostic instances for `TFC1` to the training scheme of the Duet model, this can lead to improvements in retrieval effectiveness, although our unsophisticated approach has not achieved significant improvements.

Concluding, we believe that the axiomatic approach to diagnosing neural IR models presented in this work is a step forward to gaining valuable insights into the black boxes that deep models are generally considered to be. We hope our work may prove a fruitful resource for evaluation in the field of neural IR on the road towards achieving superior performance without losing sight of a better fundamental understanding of IR.

5.2. Future Work

We are one of the first (among [98, 108]) to consider axiomatic thinking within the field of neural IR. Moreover, we have used axiomatic thinking to address a problem that so far has remained largely unexplored: the diagnosis of neural IR approaches [49]. Hence, there are many ways to build forth on the research conducted in this thesis, either improving the conducted research or extending it, as we will subsequently discuss in Section 5.2.1 and 5.2.2.

5.2.1. Improvements

We consider *improvements* as being ways to address limitations of this work within its considered ranges (of e.g. axioms, tasks and models). We discuss several of such improvements in the following paragraphs.

A different toolkit

All neural IR models considered in this work have been employed using the MatchZoo retrieval toolkit. MatchZoo—specifically, the MatchZoo version 1.0 used in this work—is however known to have several issues[1]. A multitude of these issues have been closed without being addressed, since they were raised with regard to version 1.0, which is no longer supported as version 2.0 has recently been released. It would therefore be interesting to see if the reported results can be reproduced with a different deep net toolkit (or separate implementations of neural models. Beyond a reproduction of results through MatchZoo 2.0, it would be interesting to test the proposed methodology on the same models beyond the MatchZoo toolkit, for example by employing the baseline implementations used in [2][2]. This would allow one to strengthen or refute conclusions drawn in this work based upon experiments in which we have relied on MatchZoo.

Constraining document length differences

In the main results of the axiomatic diagnoses conducted in this work (i.e. Table 4.5 and Table 4.6), we have not taken into account the differences in length of documents within diagnostic instances. As a result, we deviate from the original axioms that assumed documents to be of exact equal length. Instead of not constraining document length differences at all, we have researched a parametric relaxation (i.e. constraining document length differences in included diagnostic instances to be no larger than a certain (relative) parameter value). However, we have not been able to identify a value that is specifically suitable for this parameter. Future work may further research methods for constraining document length differences, for example by considering different parametric constraints (e.g. a parametric relaxation with axiom-dependent values).

Validity of axioms

We have also briefly researched to what extent axioms are valid heuristics in the employed corpora. We found that diagnostic instances sometimes test whether a model has (according to some axiom) "correctly" ranked a non-relevant documents above a relevant document, especially if unconstrained document length relaxation is applied. We also studied the ratio of such diagnostic instances compared to diagnostic instances that do test whether a model ranked a relevant document above a non-relevant document instances. This ratio seems to be a related[3] to the extent to which diagnostic instances can predict/explain the retrieval effectiveness of a model. Future work can propose more sophisticated approaches to determine to what extent axioms describe valid heuristics in (modern) corpora. For example, beyond looking at how often the axioms rank a relevant answer above a non-relevant answer and vice-versa as done in our work, one could investigate upper bounds on axiomatic performance. For instance, one could employ an omniscient oracle ranking (i.e. a ranking that puts all relevant documents on top) and obtain the maximum axiomatic performance for each axiom (and the axioms combined). If this upper bound is low, it could indicate that the axiom does not indicate a relevant heuristic for the considered dataset. This is in similar spirit as a strategy designed by Aslam and Montague [8] for obtaining upper bounds on the performance of metasearch models. They researched various models that combine ranked lists of documents returned by multiple search engines in response to a given query, so that

[1] See https://github.com/NTMC-Community/MatchZoo/issues.

[2] See https://github.com/wasiahmad/mnsrf_ranking_suggestion.

[3] We carefully adopt the word "seems to be related" as the majority of diagnostic instances we use in our diagnoses consider two non-relevant documents and we only looked at the results for three axioms on two datasets.

the performance of the combination is optimized. For obtaining upper bounds on axioms one could then use axioms (instead of search engines) to obtain individual (partial) ranked results to combine in the final ranking, e.g. under the axiomatic re-ranking strategy proposed by Hagen et al. [53] (taking a random ranking of the candidate documents for a query as input to re-rank).

A discriminative metric

Another improvement to our work could be the inclusion of a specific metric for axiomatic performance. In this work, we simply considered the fraction of fulfilled instances per axiom. Such a simple metric however has limitations. For instance, it does not incorporate how far a model is off, whereas it is desired that larger errors should be penalized more than smaller errors [74]. For example, if an axiom prescribes that $S(\boldsymbol{q},\boldsymbol{d_i}) > S(\boldsymbol{q},\boldsymbol{d_j})$, while for a certain model $\boldsymbol{d_i}$ is assigned a *slightly* lower score than $\boldsymbol{d_j}$ for a query $\boldsymbol{q}$ this should be penalized less than when the difference is (much) larger). Moreover, the metric adopted in this work does not allow us to incorporate how important we deem a certain diagnostic instance. A weight of importance could for instance be assigned based on the relevance labels of documents or length differences between them. For example, "wrongly" ranking a relevant document $\boldsymbol{d_j}$ above a non-relevant document $\boldsymbol{d_i}$ should intuitively be penalized less than wrongly ranking a non-relevant document $\boldsymbol{d_j}$ above a relevant document $\boldsymbol{d_i}$). Incorporating such desiderata in a metric for axiomatic performance would be an interesting direction for future work.

Improving retrieval effectiveness

In this work we have enriched training data of a neural model with diagnostic instances with the aim of improving its axiomatic performance and retrieval effectiveness. We have however only briefly researched this strategy, which has not lead to significant improvements in retrieval effectiveness. Future work may further develop approaches that can improve the retrieval effectiveness of deep nets based on axiomatic thinking. This can be done through conducting more experiments on the proposed training data augmentation with diagnostic instances (e.g. considering pre-training (like [35]) or various distributions of training instances and diagnostic instances). Future work can also move towards the work of Rosset et al. [108] by directly incorporating axiomatic knowledge in the loss function. For instance, we can transform the regular ranked hinge loss function to incorporate a regularization term for each axiom. These terms are then activated if the considered instance fulfills the conditions of an axiom and can for instance be weighted (e.g. based upon the document length difference between the two documents in the instance or to what extent the axiom is a valid heuristic in a dataset). Although this approach generally follows the methodology proposed in [108], it differs from that approach as it would not require the perturbation of documents, as it employs diagnostic instances that have been obtained with converted axioms (assuming we would consider axioms for which this is possible, e.g. not $\overline{\texttt{LNC2}}$).

5.2.2. Extensions

We consider *extensions* to be continuations of this work beyond the studied ranges. We discuss several of such extensions in the following paragraphs.

More axioms

The first extension that comes to mind is the inclusion of additional axioms for diagnosis. We have included axioms on the notions of term frequency ($\overline{\texttt{TFC1}}$, $\overline{\texttt{TFC2}}$), inverse document frequency ($\overline{\texttt{M-TDC}}$) and document length normalization ($\overline{\texttt{LNC2}}$). To obtain a more complete diagnosis of these notions, one could evidently include the TFC3, LNC1 and TF-LNC axioms, that were proposed together with the axioms covered in this work in [42, 43]. However, these works diagnosed traditional *bag-of-words* models such as BM25 and QL, that do not consider notions such as semantic similarity or proximity in their retrieval formula. Hence, a diagnosis that includes these seven axioms, can never diagnose the effectiveness of models on e.g. documents that contain no query term (to give a concrete example, 5-12% of the relevant documents contained in TREC-2, TREC-6, TREC-7 and TREC-2005 do not contain a single query term [130]). Moreover, novel axioms can be proposed to test even more aspects of IR models through axiomatic diagnoses.

The inclusion of additional axioms may however result in contradicting axioms, for instance, $\overline{\texttt{TFC1}}$ may prescribe to rank $\boldsymbol{d_1}$ above $\boldsymbol{d_2}$ since $\boldsymbol{d_1}$ has a (slightly) larger count of query terms compared to $\boldsymbol{d_2}$, whereas $\boldsymbol{d_2}$ may very well have (much) larger count of terms that are semantically similar to the query terms and therefore a semantic axiom may prescribe to rank $\boldsymbol{d_2}$ above $\boldsymbol{d_1}$. Solutions that could be researched in future work include removing contradicting instances from diagnostic datasets or adapting axioms (e.g. by additional constraints, like we did for $\overline{\texttt{M-TDC}}$ in Section 3.3.4) so that contradictions do not occur. The inclusion of more existing axioms and newly created axioms can enrich the axiomatic diagnosis and provide a better explanation for the performance of IR models.

More tasks, datasets and models

The second extension that comes to mind is the inclusion of additional tasks, datasets and models. Such an extension can test the generalizability of the methodologies proposed in this work. Whereas the traditional axiomatic approaches were mostly validated on ad-hoc retrieval tasks [39, 41–43, 78], we have researched two re-ranking datasets one of which encompasses a QA task and the other an ad-hoc retrieval task. Many other IR tasks such as collaborative filtering, key term extraction and definition finding, can also be defined as a ranking problem [76] and may hence be studied using axioms on ranking like our work (although the validity of the heuristics encapsulated in axioms may differ across different tasks, as discussed in Section 5.2.1). Next to employing different datasets and tasks, one may also consider studying more retrieval models. As was also discusses in Section 5.2.1, different repositories may be used to this end, but another approach is to use available model outputs for a diagnosis instead of re-running models to obtain this output, for example by using past TREC results[4]. However, the use of such outputs does not guarantee that the models that produced the results have received the same input nor that the pre-processing can be reproduced. The diagnostic dataset creation pipeline proposed in this work required the pre-processing of the data from which we sample instances must be the same. A problem that evidently can also be studied. One solution would be to instead feed diagnostic instances to a (trained) model to diagnose its the fulfillment of diagnostic instances. Although the model may have been trained on data that has been pre-processed differently, this approach does allow us to ensure that the conditions of the proposed axioms hold in the instances upon which we base our diagnosis.

Improvements beyond retrieval effectiveness

Another direction for future studies is to try to obtain improvements of neural models based on axiomatic thinking that go beyond retrieval effectiveness. The focus of this work has been to study neural IR models by measuring the axiomatic performance as a potential indicator for performance as measured in retrieval effectiveness. However, improvements of neural IR models may provide improvements that go beyond retrieval effectiveness, such as training time as obtained in e.g. [108]. This may be realized, for example, in similar spirit as the pre-training conducted in [35], following the curriculum learning approach [11, 54]: first training models on diagnostic instances (i.e. to learn basic retrieval heuristics) and subsequently fine tuning them on instances from a regular corpus.

Diagnosing datasets

An axiomatic analysis of answers in a dataset, may be used as a tool to reveal “issues” in existing datasets, by diagnosing the difficulty of novel and existing datasets. For example, if we consider instances from an existing dataset and can find that one axiom (nearly) always ranks a relevant document above a non-relevant, it may identify a heuristic that can obtain optimal performance on this dataset - although this is a very extreme example and more sophisticated methods may be created to identify more subtle patterns. This is in similar spirit as the results [68] achieved on the bAbI tasks [129] (which we introduced in Section 2.4.2). Such biases have also been found in other datasets not studied by Kaushik and Lipton [68] from the NLP and Computer Vision communities [68]. Note that such findings may also be obtained from diagnoses, i.e. if a neural model mimics the one axiom in the previous example and precisely follows its heuristic after training. Identifying such issues is especially important in the neural domain, as issues may go unnoticed in neural models that have inner-workings that are difficult to interpret.

[4] See https://trec.nist.gov/results.html.

Bibliography

[1] N. Abdul-jaleel, J. Allan, W.B. Croft, O. Diaz, L. Larkey, Xiaoyan Li, M Smucker, and C. Wade. Umass at trec 2004: Novelty and hard. In *In Proceedings of TREC-13*, 2004.

[2] W.U. Ahmad, K.-W Chang, and H. Wang. Multi-task learning for document ranking and query suggestion. In *The International Conference on Machine Learning*, 2018.

[3] M. Almasri, C. Berrut, and J.-P. Chevallet. A comparison of deep learning based query expansion with pseudo-relevance feedback and mutual information. In *European conference on information retrieval*, pages 709–715. Springer, 2016.

[4] E. Amigó, J. Gonzalo, and F. Verdejo. A general evaluation measure for document organization tasks. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 643–652. ACM, 2013.

[5] E. Amigo, H. Fang, S. Mizzaro, and C. Zhai. Axiomatic Thinking for Information Retrieval: And Related Tasks. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1419–1420, 2017.

[6] M. Ariannezhad, A. Montazeralghaem, H. Zamani, and A. Shakery. Improving Retrieval Performance for Verbose Queries via Axiomatic Analysis of Term Discrimination Heuristic. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1201–1204, 2017.

[7] T. G. Armstrong, A. Moffat, W. Webber, and J. Zobel. Improvements that don't add up: ad-hoc retrieval results since 1998. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 601–610, 2009.

[8] J. A. Aslam and M. Montague. Models for metasearch. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 276–284. ACM, 2001.

[9] R. Baeza-Yates, B.A.N. Ribeiro, et al. *Modern information retrieval.* New York: ACM Press; Harlow, England: Addison-Wesley,, 2011.

[10] J.P. Ballantine and A.R. Jerbert. Distance from a line, or plane, to a poin. *The American Mathematical Monthly*, 59(4):242–243, 1952.

[11] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM, 2009.

[12] F. Borisyuk, K. Kenthapadi, D. Stein, and B. Zhao. Casmos: A framework for learning candidate selection models over structured queries and documents. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 441–450. ACM, 2016.

[13] G.-I. Brokos. Document reranking with deep learning in information retrieval. Master's thesis, Athens University of Economics and Business, 2018.

[14] O. Chapelle, D. Metlzer, Y. Zhang, and P. Grinspan. Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 621–630, New York, NY, USA, 2009. ACM.

[15] H. Chen, F. X. Han, D. Niu, D. Liu, K. Lai, C. Wu, and Y. Xu. MIX: Multi-Channel Information Crossing for Text Matching. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 110–119, 2018.

[16] C.L. Clarke, N. Craswell, and E.M. Voorhees. Overview of the trec 2012 web track. Technical report, National Institute of Standards and Technology, Gaithersburg MD, 2012.

[17] C. Cleverdon. The cranfield tests on index language devices. In *Aslib proceedings*, volume 19, pages 173–194. MCB UP Ltd, 1967.

[18] S. Clinchant and E. Gaussier. Is document frequency important for PRF? In *Conference on the Theory of Information Retrieval*, pages 89–100. Springer, 2011.

[19] S. Clinchant and E. Gaussier. A theoretical analysis of pseudo-relevance feedback models. In *Proceedings of the 2013 Conference on the Theory of Information Retrieval*, page 6, 2013.

[20] D. Cohen, J. Foley, H. Zamani, J. Allan, and W.B. Croft. Universal approximation functions for fast learning to rank: Replacing expensive regression forests with simple feed-forward networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1017–1020. ACM, 2018.

[21] D. Cohen, B. O'Connor, and W. B. Croft. Understanding the Representational Power of Neural Retrieval Models Using NLP Tasks. In *Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval*, ICTIR '18, pages 67–74, 2018.

[22] D. Cohen, L. Yang, and W.B. Croft. WikiPassageQA: A Benchmark Collection for Research on Non-factoid Answer Passage Retrieval. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, pages 1165–1168, 2018.

[23] F. Colace, M. De Santo, L. Greco, and P. Napoletano. Improving relevance feedback-based query expansion by the use of a weighted word pairs approach. *Journal of the Association for Information Science and Technology*, 66(11):2223–2234, 2015.

[24] K. Collins-Thompson, P. Bennett, F. Diaz, C.L. Clarke, and E.M. Voorhees. Trec 2013 web track overview. 2014.

[25] K. Collins-Thompson, C. Macdonald, P. Bennett, F. Diaz, and E.M. Voorhees. Trec 2014 web track overview. Technical report, Michigan University, Ann Arbor, 2015.

[26] M. Crane. Questionable answers in question answering research: Reproducibility and variability of published results. *Transactions of the Association of Computational Linguistics*, 6:241–252, 2018.

[27] N. Craswell, D. Fetterly, and M. Najork. Microsoft research at trec 2010 web track. In *TREC*, 2010.

[28] N. Craswell, W.B. Croft, M. de Rijke, J. Guo, and B. Mitra. SIGIR 2017 Workshop on Neural Information Retrieval (Neu-IR'17). In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1431–1432, 2017.

[29] N. Craswell, W.B Croft, M. de Rijke, J. Guo, and B. Mitra. Report on the Second SIGIR Workshop on Neural Information Retrieval (Neu-IR'17). In *ACM SIGIR Forum*, volume 51, pages 152–158, 2018.

[30] J.S. Culpepper, F. Diaz, and M.D. Smucker. Research frontiers in information retrieval: Report from the third strategic workshop on information retrieval in lorne (swirl 2018). In *ACM SIGIR Forum*, volume 52, pages 34–90. ACM, 2018.

[31] Z. Dai, C. Xiong, J. Callan, and Z. Liu. Convolutional neural networks for soft-matching n-grams in ad-hoc search. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pages 126–134. ACM, 2018.

[32] R. Datta, D. Joshi, J. Li, and J.Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys (Csur)*, 40(2):5, 2008.

[33] C. De Boom, S. Van Canneyt, T. Demeester, and B. Dhoedt. Representation learning for very short texts using weighted word embedding aggregation. *Pattern Recognition Letters*, 80:150–156, 2016.

[34] S. De Cnudde, D. Martens, F. Provost, et al. An exploratory study towards applying and demystifying deep learning classification on behavioral big data. Technical report, 2018.

[35] M. Dehghani, H. Zamani, A. Severyn, J. Kamps, and W.B. Croft. Neural ranking models with weak supervision. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 65–74. ACM, 2017.

[36] F. Diaz and D. Metzler. Improving the estimation of relevance models using large external corpora. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 154–161. ACM, 2006.

[37] L. Dietz, M. Verma, F. Radlinski, and N. Craswell. Trec complex answer retrieval overview. In *Proceedings of TREC*, 2017.

[38] Y. Fan, L. Pang, J. Hou, J. Guo, Y. Lan, and X. Cheng. MatchZoo: A Toolkit for Deep Text Matching. *arXiv preprint arXiv:1707.07270*, 2017.

[39] H. Fang. A re-examination of query expansion using lexical resources. *proceedings of ACL-08: HLT*, pages 139–147, 2008.

[40] H. Fang and C. Zhai. An exploration of axiomatic approaches to information retrieval. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 480–487, 2005.

[41] H. Fang and C. Zhai. Semantic term matching in axiomatic approaches to information retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 115–122, 2006.

[42] H. Fang, T. Tao, and C. Zhai. A formal study of information retrieval heuristics. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 49–56, 2004.

[43] H. Fang, T. Tao, and C. Zhai. Diagnostic evaluation of information retrieval models. *ACM Transactions on Information Systems (TOIS)*, 29(2):7, 2011.

[44] N. Fuhr. Optimum polynomial retrieval functions based on the probability ranking principle. *ACM Transactions on Information Systems (TOIS)*, 7(3):183–204, 1989.

[45] K. Fukushima. Neural network model for a mechanism of pattern recognition unaffected by shift in position-neocognitron. *IEICE Technical Report, A*, 62(10):658–665, 1979.

[46] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980.

[47] A. Gonzalez, I. Augenstein, and A. Søgaard. A strong baseline for question relevancy ranking. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4810–4815, 2018.

[48] J. Guo, Y. Fan, Q. Ai, and W.B. Croft. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 55–64, 2016.

[49] J. Guo, Y. Fan, L. Pang, L. Yang, Q. Ai, H. Zamani, C. Wu, W.B. Croft, and X. Cheng. A deep look into neural ranking models for information retrieval. *arXiv preprint arXiv:1903.06902*, 2019.

[50] Y. Gupta, A. Saini, and A.K. Saxena. A review on important aspects of information retrieval.

[51] I. Guy. Searching by talking: Analysis of voice queries on mobile web search. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 35–44. ACM, 2016.

[52] M. Hagen. Axiomatic result re-ranking. 2017.

[53] M. Hagen, M. Völske, S. Göring, and B. Stein. Axiomatic result re-ranking. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 721–730, 2016.

[54] C. Hauff. Neural ir. Delft University of Technology, Information Retrieval, IN4325, 2018.

[55] H. Hazimeh and C. Zhai. Axiomatic analysis of smoothing methods in language models for pseudo-relevance feedback. In *Proceedings of the 2015 international conference on the theory of information retrieval*, pages 141–150. ACM, 2015.

[56] F. Hill, A. Bordes, S. Chopra, and J. Weston. The goldilocks principle: Reading children's books with explicit memory representations. *arXiv preprint arXiv:1511.02301*, 2015.

[57] K. Hofmann et al. Fast and reliable online learning to rank for information retrieval. In *SIGIR Forum*, volume 47, page 140, 2013.

[58] B. Hu, Z. Lu, H. Li, and Q. Chen. Convolutional neural network architectures for matching natural language sentences. In *Advances in neural information processing systems*, pages 2042–2050, 2014.

[59] K. Hui, A. Yates, K. Berberich, and G. de Melo. Pacrr: A position-aware neural ir model for relevance matching. *arXiv preprint arXiv:1704.03940*, 2017.

[60] K. Hui, A. Yates, K. Berberich, and G. de Melo. Co-PACRR: A Context-Aware Neural IR Model for Ad-hoc Retrieval. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining. WSDM*, volume 18, page 2, 2018.

[61] A.G. Ivakhnenko. The group method of data of handling; a rival of the method of stochastic approximation. *Soviet Automatic Control*, 13:43–55, 1968.

[62] A.G. Ivakhnenko. Polynomial theory of complex systems. *IEEE transactions on Systems, Man, and Cybernetics*, (4):364–378, 1971.

[63] A.G. Ivakhnenko and V.G. Lapa. *Cybernetic predicting devices*. CCM Information Corporation, 1965.

[64] A.G. Ivakhnenko and V.G. Lapa. *Cybernetics and Forecasting Techniques Modern Analytic and Computational Method in Science and Mathematics*. New York: American Elsevier Publishing Company, Inc, 1967.

[65] R. Jia and P. Liang. Adversarial Examples for Evaluating Reading Comprehension Systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, 2017.

[66] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C.L. Zitnick, and R. Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 1988–1997. IEEE, 2017.

[67] M. Karimzadehgan and C. Zhai. Axiomatic analysis of translation language model for information retrieval. In *European Conference on Information Retrieval*, pages 268–280. Springer, 2012.

[68] D. Kaushik and Z. C. Lipton. How much reading does reading comprehension require? a critical investigation of popular benchmarks. *arXiv preprint arXiv:1808.04926*, 2018.

[69] O. Kolomiyets and M.-F. Moens. A survey on question answering technology from an information retrieval perspective. *Information Sciences*, 181(24):5412–5434, 2011.

[70] V. Lavrenko and W.B. Croft. Relevance-based language models. 2001.

[71] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436, 2015.

[72] W. Li, B. Kan, and W. C. Mak. Recurrent neural network language model adaptation derived document vector. *CoRR*, abs/1611.00196, 2016.

[73] J. Lin. The neural hype and comparisons against weak baselines. In *ACM SIGIR Forum*, volume 52, pages 40–51. ACM, 2019.

[74] C. Lioma, J. Simonsen, and B. Larsen. Evaluation measures for relevance and credibility in ranked lists. In *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*, pages 91–98. ACM, 2017.

[75] B. Liu, X. Lu, O. Kurland, and J.S. Culpepper. Improving search effectiveness with field-based relevance modeling. In *Proceedings of the 23rd Australasian Document Computing Symposium*, page 11. ACM, 2018.

[76] T.-Y. Liu. *Learning to rank for information retrieval.* Springer Science & Business Media, 2011.

[77] Y. Lv and C. Zhai. A comparative study of methods for estimating query language models with pseudo feedback. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1895–1898. ACM, 2009.

[78] Y. Lv and C. Zhai. Lower-bounding term frequency normalization. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 7–16, 2011.

[79] D.J.C. MacKay and L.C.B. Peto. A hierarchical dirichlet language model. *Natural language engineering*, 1(3):289–308, 1995.

[80] C. Manning, P. Raghavan, and H. Schütze. Classical and web information retrieval systems: algorithms, mathematical foundations and practical issues in. *Introduction to information retrieval, Cambridge*, 2008.

[81] G. Marcus. Deep learning: A critical appraisal. *arXiv preprint arXiv:1801.00631*, 2018.

[82] M.E. Maron and J.L. Kuhns. On relevance, probabilistic indexing and information retrieval. *Journal of the ACM (JACM)*, 7(3):216–244, 1960.

[83] R. McDonald, G.-I. Brokos, and I. Androutsopoulos. Deep relevance ranking using enhanced document-query interactions. *arXiv preprint arXiv:1809.01682*, 2018.

[84] J. Miao, J.X. Huang, and Z. Ye. Proximity-based rocchio's model for pseudo relevance. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 535–544. ACM, 2012.

[85] B. Mitra and N. Craswell. Neural Models for Information Retrieval. *arXiv preprint arXiv:1705.01509*, 2017.

[86] B. Mitra and N. Craswell. An introduction to neural information retrieval. *Foundations and Trends in Information Retrieval (to appear)*, 2018.

[87] B. Mitra, F. Diaz, and N. Craswell. Learning to match using local and distributed representations of text for web search. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1291–1299. International World Wide Web Conferences Steering Committee, 2017.

[88] A. Moffat. Seven numeric properties of effectiveness metrics. In *Asia Information Retrieval Symposium*, pages 1–12. Springer, 2013.

[89] A. Montazeralghaem, H. Zamani, and A. Shakery. Axiomatic analysis for improving the log-logistic feedback model. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 765–768, 2016.

[90] S.-H. Na. Two-stage document length normalization for information retrieval. *ACM Transactions on Information Systems (TOIS)*, 33(2):8, 2015.

[91] F. Nanni, B. Mitra, M. Magnusson, and L. Dietz. Benchmark for complex answer retrieval. In *Proceedings of the ACM SIGIR international conference on theory of information retrieval*, pages 293–296. ACM, 2017.

[92] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*, 2016.

[93] Y. Nie, Y. Li, and J.-Y. Nie. Empirical Study of Multi-level Convolution Models for IR Based on Representations and Interactions. In *Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval*, ICTIR '18, pages 59–66, 2018.

[94] R. Nogueira and K. Cho. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*, 2019.

[95] K.D. Onal, Y. Zhang, I.S. Altingovde, M.M. Rahman, P. Karagoz, A. Braylan, B. Dang, H.-L. Chang, H. Kim, Q. McNamara, et al. Neural information retrieval: At the end of the early years. *Information Retrieval Journal*, 21(2-3):111–182, 2018.

[96] L. Pang, Y. Lan, J. Guo, J. Xu, S. Wan, and X. Cheng. Text Matching as Image Recognition. In *AAAI*, pages 2793–2799, 2016.

[97] L. Pang, Y. Lan, J. Guo, Jun Xu, and X. Cheng. A study of matchpyramid models on ad-hoc retrieval. *arXiv preprint arXiv:1606.04648*, 2016.

[98] L. Pang, Y. Lan, J. Guo, J. Xu, and X. Cheng. A deep investigation of deep ir models. *arXiv preprint arXiv:1707.07700*, 2017.

[99] L. Pang, Y. Lan, J. Guo, J. Xu, J. Xu, and X. Cheng. Deeprank: A new deep architecture for relevance ranking in information retrieval. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 257–266, 2017.

[100] J. Parapar and A. Barreiro. Promoting divergent terms in the estimation of relevance models. In *Conference on the Theory of Information Retrieval*, pages 77–88. Springer, 2011.

[101] T. Qin, T.-Y. Liu, J. Xu, and H. Li. Letor: A benchmark collection for research on learning to rank for information retrieval. *Information Retrieval*, 13(4):346–374, 2010.

[102] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[103] J. Rao, W. Yang, Y. Zhang, F. Ture, and J. Lin. Multi-Perspective Relevance Matching with Hierarchical ConvNets for Social Media Search. *arXiv preprint arXiv:1805.08159*, 2018.

[104] D. Rennings, F. Moraes, and C. Hauff. An axiomatic approach to diagnosing neural ir models. In *Advances in Information Retrieval*, pages 489–503. Springer International Publishing, 2019. ISBN 978-3-030-15712-8.

[105] S.E. Robertson. The probability ranking principle in ir. *Journal of documentation*, 33(4):294–304, 1977.

[106] S.E. Robertson and K.S. Jones. Relevance weighting of search terms. *Journal of the American Society for Information science*, 27(3):129–146, 1976.

[107] S.E. Robertson, S. Walker, S. Jones, M.M. Hancock-Beaulieu, M. Gatford, et al. Okapi at trec-3. *Nist Special Publication Sp*, 109:109, 1995.

[108] C. Rosset, B. Mitra, C. Xiong, N. Craswell, X. Song, and S. Tiwary. An axiomatic approach to regularizing neural ranking models. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval (to appear)*. ACM, April 2019.

[109] R.K. Saha, M. Lease, S. Khurshid, and D.E. Perry. Improving bug localization using structured information retrieval. In *2013 28th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 345–355. IEEE, 2013.

[110] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[111] M. Sanderson and W.B. Croft. The history of information retrieval research. *Proceedings of the IEEE*, 100(Special Centennial Issue):1444–1451, 2012.

[112] M. Sanderson et al. Test collection based evaluation of information retrieval systems. *Foundations and Trends® in Information Retrieval*, 4(4):247–375, 2010.

[113] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[114] S. Shi, J.-R. Wen, Q. Yu, R. Song, and W.-Y. Ma. Gravitation-based model for information retrieval. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 488–495, 2005.

[115] I. Soboroff. A comparison of pooled and sampled relevance judgments in the trec 2006 terabyte track. In *EVIA@ NTCIR*, 2007.

[116] I. Soboroff, N. Craswell, C. L. Clarke, and G. Cormack. Overview of the trec 2011 web track. Technical report, 2011.

[117] M. Sokolova. Assessing invariance properties of evaluation measures. In *Proceedings of the Workshop on Testing of Deployable Learning and Decision Systems, the 19th Neural Information Processing Systems Conference (NIPS 2006)*, 2006.

[118] K. Spark-Jones. Report on the need for and provision of an'ideal'information retrieval test collection. *Computer Laboratory*, 1975.

[119] T. Strohman, D. Metzler, H. Turtle, and W.B. Croft. Indri: A language model-based search engine for complex queries. In *Proceedings of the International Conference on Intelligent Analysis*, volume 2, pages 2–6, 2005.

[120] Z. Tang and G.H. Yang. Deeptilebars: Visualizing term distribution for neural information retrieval. *arXiv preprint arXiv:1811.00606*, 2018.

[121] T. Tao and C. Zhai. An exploration of proximity measures in information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 295–302, 2007.

[122] S. Teufel. An overview of evaluation methods in trec ad hoc information retrieval and trec question answering. In *Evaluation of text and speech systems*, pages 163–186. Springer, 2007.

[123] C. Van Gysel, M. de Rijke, and E. Kanoulas. Semantic entity retrieval toolkit. *arXiv preprint arXiv:1706.03757*, 2017.

[124] E.M. Voorhees. The philosophy of information retrieval evaluation. In *Workshop of the cross-language evaluation forum for european languages*, pages 355–370. Springer, 2001.

[125] E.M. Voorhees. The trec robust retrieval track. In *ACM SIGIR Forum*, volume 39, pages 11–20. ACM, 2005.

[126] S. Wan, Y. Lan, J. Guo, J. Xu, L. Pang, and X. Cheng. A Deep Architecture for Semantic Matching with Multiple Positional Sentence Representations. In *AAAI*, volume 16, pages 2835–2841, 2016.

[127] S. Wan, Y. Lan, J. Xu, J. Guo, L. Pang, and X. Cheng. Match-srnn: Modeling the recursive matching structure with spatial rnn. *arXiv preprint arXiv:1604.04378*, 2016.

[128] M. Wang, N.A. Smith, and T. Mitamura. What is the jeopardy model? a quasi-synchronous grammar for qa. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007.

[129] J. Weston, A. Bordes, S. Chopra, A.M. Rush, B. van Merriënboer, A. Joulin, and T. Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*, 2015.

[130] H.C. Wu, R.W.P. Luk, K.-F. Wong, and K.L. Kwok. A retrospective study of a hybrid document-context based retrieval model. *Information processing & management*, 43(5):1308–1331, 2007.

[131] C. Xiong, Z. Dai, J. Callan, Z. Liu, and R. Power. End-to-end neural ad-hoc ranking with kernel pooling. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 55–64. ACM, 2017.

[132] B. Xu, H. Lin, Y. Lin, L. Yang, and K. Xu. Improving pseudo-relevance feedback with neural network-based word representations. *IEEE Access*, 6:62152–62165, 2018.

[133] L. Yang, Q. Ai, J. Guo, and W.B. Croft. aNMM: Ranking short answer texts with attention-based neural matching model. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 287–296, 2016.

[134] Y. Yang, W.-T. Yih, and C. Meek. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, 2015.

[135] Z. Yang, Q. Lan, J. Guo, Y. Fan, X. Zhu, Y. Lan, Y. Wang, and X. Cheng. A Deep Top-K Relevance Matching Model for Ad-hoc Retrieval. In *China Conference on Information Retrieval*, pages 16–27, 2018.

[136] W.-T. Yih, K. Toutanova, J.C. Platt, and C. Meek. Learning discriminative projections for text similarity measures. In *Proceedings of the fifteenth conference on computational natural language learning*, pages 247–256. Association for Computational Linguistics, 2011.

[137] W. Yin and H. Schütze. Multigrancnn: An architecture for general matching of text chunks on multiple levels of granularity. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 63–73, 2015.

[138] H. Zamani and W.B. Croft. On the theory of weak supervision for information retrieval. In *Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 147–154. ACM, 2018.

[139] H. Zamani, W.B. Croft, and J.S. Culpepper. Neural query performance prediction using weak supervision from multiple signals. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 105–114. ACM, 2018.

[140] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 334–342, New York, NY, USA, 2001. ACM. ISBN 1-58113-331-6.

[141] X. Zhang, J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657, 2015.

[142] L. Zighelnic and O. Kurland. Query-drift prevention for robust query expansion. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 825–826. ACM, 2008.

[143] B. Zoph and Q.V. Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.

A

Hyper-parameter Tuning

A.1. Traditional Models on `WikiPassageQA`

An overview of the tested parameters per model employed in the tuning process conducted for the `WikiPassageQA` dataset is displayed in Table A.1.

model	**parameter**	**tested values**
BM25	k_1	$\{0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0\}$
	b	$\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$
	k_3	$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20\}$
QL	μ	$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 25, 50, 75, 100, 250, 500, 750, 1000, 2000, 3000, 4000, 5000\}$
RM3	μ	$\{100, 750, 3000\}$
	$fbDocs$	$\{5, 10, 25, 50\}$
	$fbTerms$	$\{50, 100, 500\}$
	$fbMu$	$\{100, 750, 3000\}$
	$fbOrigWeight$	$\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$

Table A.1: Parameter values per model as adopted in the parameter tuning process, all combinations of the listed parameter values were tested on the train and dev splits of the `WikiPassageQA` dataset.

A.2. Traditional Models on `MSMarco`

An overview of the tested parameters per model employed in the tuning process conducted for the `MSMarco` dataset is displayed in Table A.2.

model	**parameter**	**tested values**
BM25	k_1	$\{0.6, 0.9, 1.2, 1.5, 1.8\}$
	b	$\{0.25, 0.50, 0.75, 1.00, 1.25\}$
	k_3	$\{1, 3, 7, 10\}$
QL	μ	$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 25, 50, 75, 100, 250, 500, 750, 1000, 2000, 3000, 4000, 5000\}$
RM3	μ	$\{10, 1000, 2500, 3000\}$
	$fbDocs$	$\{5, 10, 20\}$
	$fbTerms$	$\{50, 100, 500\}$
	$fbMu$	$\{10, 100, 2500, 3000\}$
	$fbOrigWeight$	$\{0.3, 0.6, 0.9\}$

Table A.2: Parameter values per model as adopted in the parameter tuning process, all combinations of the listed parameter values were tested on the train and dev splits of the (subset of the) `MSMarco` dataset.

B

Detailed Results

B.1. Retrieval Models on `WikiPassageQA` and `MSMarco`

B.1.1. `WikiPassageQA`

An overview of the obtained retrieval effectiveness and axiomatic performances for the `WikiPassageQA` dataset rounded to four decimals of both the traditional and neural IR models employed in this work is displayed in Table B.1.

B.1.2. `MSMarco`

An overview of the obtained retrieval effectiveness and axiomatic performances for the `MSMarco` dataset rounded to four decimals of both the traditional and neural IR models employed in this work is displayed in Table B.2.

B.2. Duet with Training Data Augmentation on `WikiPassageQA`

An overview of the obtained retrieval effectiveness and axiomatic performances for the various training data augmentation strategies for Duet on the `WikiPassageQA` dataset rounded to four decimals is displayed in Table B.3.

	Retrieval effectiveness			Performance per axiom				
	MAP	MRR	P@5	$\overline{\texttt{TFC1}}$	$\overline{\texttt{TFC2}}$	$\overline{\texttt{M-TDC}}$	$\overline{\texttt{LNC2}}^{Test}$	$\overline{\texttt{LNC2}}^{All}$
Random				0.50		0.50	0.50	0.50
^{1} BM25	0.5199^{4,5,6,7}	0.5983^{4,5,6,7}	0.1821^{4,5,6}	0.7251^{4,5,6}	**0.9792**^{2,3,4,5,6,7,8,9}	**0.9962**^{2,3,4,5,6,7,8,9}	**0.7971**	**0.7971**
^{2} RM3	0.5349^{1,4,5,6,7}	0.6186^{1,4,5,6,7}	**0.1913**^{1,4,5,6}	**0.8823**^{1,3,4,5,6,7,8,9}	0.6345^{3,4,5,6,7,8,9}	0.9411^{4,5,6,7,8,9}	0.7242	0.7242
^{3} QL	**0.5355**^{1,4,5,6,7}	**0.6209**^{1,4,5,6,7}	**0.1913**^{1,4,5,6}	0.8734^{1,4,5,6,7,8,9}	0.6312^{4,5,6,7,8,9}	0.9429^{4,5,6,7,8,9}	0.6798	0.6798
^{4} Arc-I	0.1950	0.2226	0.0739	0.6828^{5}	0.5506	0.5039^{6}	0.1333	0.3924
^{5} MV-LSTM	0.2337^{4}	0.2678^{4}	0.0903^{4}	0.6763	0.5608^{4,6}	0.5092^{6}	0.1648	0.7075
^{6} Duet	0.2472^{4}	0.2877^{4}	0.0971^{4}	0.6949^{4,5}	0.5556	0.4757	0.1865	0.4689
^{7} MatchP.	0.4388^{4,5,6}	0.5088^{4,5,6}	0.1807^{4,5,6}	0.7941^{1,4,5,6}	0.5821^{4,5,6,9}	0.6253^{4,5,6}	0.0012	0.1947
^{8} DRMM	0.5597^{1,2,4,5,6}	0.6393^{1,2,3,4,5,6}	0.2024^{1,2,3,4,5,6}	0.8380^{1,4,5,6,7}	**0.5966**^{3,4,5,6,7,9}	**0.7603**^{4,5,6,7,9}	0.0513	0.1207
^{9} aNMM	**0.5734**^{1,2,3,4,5,6,7}	**0.6579**^{1,2,3,4,5,6,7}	**0.2087**^{1,2,3,4,5,6,7}	**0.8457**^{1,4,5,6,7,8}	0.5635^{4,6}	0.6882^{4,5,6,7}	**0.3770**	**0.4712**
Best in [22]	0.5608	0.6792	0.2083	?	?	?	?	?

Table B.1: Overview of models' retrieval effectiveness and fraction of fulfilled axiom instances for the `WikiPassageQA` dataset. For measuring statistical significance, we employed the Wilcoxon test and McNemar test with $p < 0.05$ on respectively measures for retrieval effectiveness and axiomatic performance.

	Retrieval effectiveness			**Performance per axiom**			
	MAP	MRR	P@5	$\overline{\texttt{TFC1}}$	$\overline{\texttt{TFC2}}$	$\overline{\texttt{M-TDC}}$	$\overline{\texttt{LNC2}}^{Test}$
Random				0.50		0.50	0.50
^{1} BM25	0.1969	0.1998	0.0586	0.5422^{4,5,6,7,8,9}	**0.5042**^{2,3,6}	**0.6927**^{2,3,4,5,6,7,8,9}	0.0024
^{2} QL	0.2090^{1}	0.2120^{1}	0.0615^{1}	**0.6757**^{1,3,4,5,6,7,8,9}	0.4322	0.6063^{3,4,5,6,7,8,9}	0.0575
^{3} RM3	**0.2125**^{1,2}	**0.2156**^{1,2}	**0.0632**^{1,2}	0.6573^{1,4,5,6,7,8,9}	0.4237	0.5872^{4,5,6,7,8,9}	0.0000
^{4} MV-LSTM	0.2485^{1,2,3}	0.2523^{1,2,3}	0.0729^{1,2,3}	0.2901^{5,7,8}	**0.5593**^{2,3,5,6,7}	**0.4618**^{6,7}	0.3113
^{5} Arc-I	0.2569^{1,2,3,4}	0.2609^{1,2,3,4}	0.0734^{1,2,3}	0.2866^{7,8}	0.4746^{4,6}	0.4584^{7,8}	0.0000
^{6} Duet	0.2826^{1,2,3,4,5}	0.2865^{1,2,3,4,5}	0.0826^{1,2,3,4,5}	0.3316^{4,5,7,8}	0.4195	0.4546^{7,8}	0.0001
^{7} DRMM	0.3186^{1,2,3,4,5,6}	0.3229^{1,2,3,4,5,6}	0.0851^{1,2,3,4,5}	0.2035^{8}	0.4703	0.3790	0.0001
^{8} aNMM	0.3197^{1,2,3,4,5,6,7}	0.3246^{1,2,3,4,5,6,7}	0.0867^{1,2,3,4,5,6}	0.1992	0.5551^{2,3,6,7}	0.3902^{7}	0.0000
^{9} MatchPyramid	**0.3288**^{1,2,3,4,5,6,7,8}	**0.3358**^{1,2,3,4,5,6,7,8}	**0.0932**^{1,2,3,4,5,6,7,8}	**0.3493**^{4,5,6,7,8}	0.4703	0.4598^{7,8}	0.0000

Table B.2: Overview of models' retrieval effectiveness and fraction of fulfilled axiom instances for the MSMarco dataset. For measuring statistical significance, we employed the Wilcoxon test and McNemar test with $p < 0.05$ on respectively measures for retrieval effectiveness and axiomatic performance.

	Retrieval effectiveness			Performance per axiom		
Training data	MAP	MRR	P@5	$\overline{\texttt{TFC1}}$	$\overline{\texttt{TFC2}}$	$\overline{\texttt{M-TDC}}$
^{1} answers	0.2472^{2}	0.2877^{2}	0.0971^{2}	0.6949	$0.5551^{2,3,4}$	0.4757
^{2} answers + $\overline{\texttt{TFC1}}_{multi\text{-}graded}$	0.1989	0.2244	0.0705	$0.7550^{1,4}$	0.4986^{4}	0.4941
^{3} answers + $\overline{\texttt{TFC1}}_{paired,1:1}$	$\mathbf{0.2690}^{2}$	$\mathbf{0.3088}^{2}$	0.1005^{2}	$0.8108^{1,2,4}$	$\mathbf{0.5439}^{2,4}$	$\mathbf{0.5143}^{1,2}$
^{4} answers + $\overline{\texttt{TFC1}}_{paired,1:1,\delta^* \leq 0.1}$	0.2615^{2}	$\mathbf{0.3109}^{2}$	$\mathbf{0.1039}^{2}$	0.7032^{1}	0.4929	$\mathbf{0.5092}^{1}$

Table B.3: Overview of retrieval effectiveness and fraction of fulfilled axiom instances for the Duet model trained on the original and augmented versions of the `WikiPassageQA` dataset. For measuring statistical significance, we employed the Wilcoxon test and McNemar test with $p < 0.05$ on respectively measures for retrieval effectiveness and axiomatic performance.