FRIEDRICH-SCHILLER-
UNIVERSITÄT
JENA

Transferring Relevance Judgments with Pairwise Preferences

BACHELOR'S THESIS

to attain the academic degree

Bachelor of Science (B.Sc.)

in Computer Science (Informatik)

FRIEDRICH SCHILLER UNIVERSITY JENA

Faculty of Mathematics and Computer Science

submitted by Fabian Hofer

born on 22.08.2002 in Eisenach

1. Referee: Prof. Dr. Matthias Hagen

2. Referee: Maik Fröbe

Jena, 13.03.2025

Abstract

Relevance judgments are essential for evaluating and comparing the effectiveness of information retrieval systems. Traditionally, human assessors manually review query-document pairs to determine relevance judgments. This process is costly, time-consuming, and must be repeated for each new test collection. This thesis addresses this challenge by presenting a method for automatically generating relevance judgments using existing annotated datasets. The proposed approach transfers relevance information from a well-judged source corpus to a subset of documents in a target corpus, enriching it with newly generated judgments. The method employs a pairwise preference approach, where a large language model compares already judged documents from the source corpus with candidate documents from the target corpus. To do this, the model is prompted to determine whether a target document is as relevant to a given query as an already judged source document, resulting in an automatically generated set of relevance judgments for the target corpus. To evaluate the effectiveness of the developed approach, the transfer method is applied to multiple existing test collections that already contain relevance judgments. Each collection serves as both a source and its own target corpus, enabling automatic evaluation by comparing the newly generated judgments with the original ones. Additionally, the approach is tested on `ClueWeb22/b` as an unjudged target corpus. By leveraging pairwise preference with already judged documents, this approach has the potential to significantly reduce the effort for manual annotation while maintaining high-quality relevance judgments, making scalable enrichment of target corpora possible.

Kurzfassung

Relevanzbewertungen sind für die Evaluierung und den Vergleich der Effektivität von Information Retrieval Systemen von entscheidender Bedeutung. Traditionell werden diese Bewertungen von menschlichen Annotatoren manuell durch die Bewertung von Anfrage-Dokument-Paaren bestimmt. Dieser Prozess ist jedoch zeitaufwändig und muss für jeden neuen Testdatensatz wiederholt werden. Diese Arbeit untersucht, ob der Aufwand für die Erstellung von Relevanzbewertungen reduziert werden kann, indem bestehende annotierte Datensätze zur automatischen Generierung neuer Bewertungen verwendet werden. Dazu wird ein Ansatz vorgestellt, der Relevanzinformationen aus einem gut annotierten Datensatz extrahiert und auf einen neuen Datensatz überträgt. Das Verfahren basiert auf paarweisen Vergleichen. Dazu werden bereits annotierte Dokumente eines Quelldatensatzes mit Dokumenten eines Zieldatensatzes in einem Large Language Model verglichen. Das Modell hat die Aufgabe zu bestimmen, ob ein Zieldokument für eine bestimmte Anfrage genauso relevant ist wie ein bereits bewertetes Quelldokument. Das Ergebnis ist eine automatisch generierte Menge von Relevanzbewertungen für den Zieldatensatz. Zur Evaluierung des Ansatzes wird dieser auf mehrere bestehende Testdatensätze angewendet, die bereits Relevanzbewertungen enthalten. Jede dieser Sammlungen dient sowohl als Quell- als auch als Zieldatensatz, was einen Vergleich der generierten Relevanzbewertungen mit den ursprünglichen Relevanzbewertungen ermöglicht. Zusätzlich wird der Transfer mit `ClueWeb22/b` als Zieldatensatz getestet. Durch die Nutzung bereits annotierter Dokumente und paarweiser Vergleiche hat dieser Ansatz das Potenzial, den manuellen Annotationsaufwand neuer Datensätze deutlich zu reduzieren und gleichzeitig qualitativ hochwertige Relevanzbewertungen zu generieren.

Contents

Chapter 1

Introduction

Static test collections, which consist of information needs, documents, and relevance judgments, are commonly used to evaluate the retrieval effectiveness of retrieval systems [26]. Retrieval effectiveness measures how well a retrieval system retrieves relevant documents to a given information need. To assess this effectiveness, relevance judgments are required to classify the retrieved results as relevant or non-relevant to an information need. However, the use of such static test collections relies on several simplications that are not realistic in practice. One assumption is that information needs, documents, and relevance judgments remain unchanged over time. While information needs are often stable, documents frequently change over time [5]. This presents a challenge for evaluation methods based on static collections, as relevance judgments made for one version of a document may not apply to later versions.

Relevance judgments are assessments made by humans as to whether a document is relevant to a given query. This process is time-consuming and expensive due to human needs. One of the first widely used test collections, the Cranfield test collection [7], included a small corpus of just 1 400 scientific abstracts with 225 queries and manually assigned relevance judgments. While feasible for small collections, judging with human effort becomes impractical when dealing with modern large-scale corpora containing millions or even billions of documents. Due to the large amount of data, only a small subset of documents can be judged anyway.

In this thesis, I propose an approach for transferring relevance judgments from an existing test collection to a target corpus using pairwise preferences. The goal is to automate the creation of relevance judgments for the target corpus, thereby enriching its set of relevance judgments and enabling more accurate evaluation of retrieval systems on the target corpus. Additionally, this approach aims to reduce the need for human judgment, which would otherwise be required for manually assessing relevance judgments.

The transfer pipeline consists of several steps that process the source and target datasets to perform pairwise preference comparisons between documents from both corpora in order to generate new relevance judgments. The pipeline starts with a source dataset that includes a document corpus and an associated retrieval task that provides a set of queries and relevance judgments for the documents in the corpus.

Only documents from the source corpus that have at least one relevance judgment contain information that can be used for relevance transfer. Therefore, an initial document selection is performed on the source document corpus. To enable a more fine-granular comparison, the selected documents are segmented into smaller text passages. This segmentation is necessary because later steps, particularly the pairwise preference evaluation, rely on transformer models, which have a maximum context length they can process. Another advantage of segmenting documents is that relevant information is typically concentrated in specific parts of a document rather than being spread throughout. To identify these relevant parts, the resulting passages are ranked to determine the most relevant passages for each query in the retrieval task. The highest-scoring passages are then used in the next step to identify documents from the target corpus that are likely to be relevant to the query and should therefore be judged. The same segmentation approach is applied to the selected candidate documents from the target corpus.

To finally conduct the pairwise preference comparisions, each target passage is paired with the most relevant passages from the source dataset for the corresponding query. A large language model is provided with tuples of the selected canidates, consisting of `(query, known source passage, target passage to judged)`. The model is prompted to determine the relevance of the target passage to the query based on the known passage. The resulting relevance scores from these pairwise comparisons are then used to generate the final relevance judgments for the target documents. After aggregating the relevance scores, the pipeline outputs a set of relevance judgments for each query in the retrieval task for the selected subset of target documents.

The final part of this thesis focuses on evaluating the effectiveness of the relevance transfer pipeline. For this purpose, widely used datasets such as `Args.me`, `Disks 4+5`, and `MS MARCO` serve as source corpora. Their existing retrieval tasks provide the foundation for the transfer process, which is evaluated in two phases. First, the pipeline is applied within the source datasets themselves to assess how accurately the transfer process can reproduce known relevance judgments. To compare the generated relevance judgments with the actual ones, rank correlation is computed between both label sets. This allows for an automated evaluation against the existing relevance judgments of the source datasets. In the second phase, the pipeline is used to transfer relevance judgments from the source datasets to `ClueWeb22/b`, the selected target corpus. Since this corpus lacks pre-existing relevance judgments, evaluation in this case requires manual assessment. To address this, a representative subset of relevance judgments will be manually assessed for `ClueWeb22/b` in order to evaluate the quality of the automatically created ones.

Chapter 2

Related Work

The idea of transferring relevance judgments from one dataset to another has been explored in multiple studies. Fröbe et al. [11] investigated the issue of near-duplicate documents within widely-used web crawls such as `ClueWeb` and `Common Crawl`. The authors proposed a deduplication approach to address this issue and, as an extension, explored the potential of transferring relevance judgments between near-duplicate documents. However, their study also emphasized that newly collected judgments remain necessary for evaluating retrieval systems on newer datasets. An example of ineffective relevance judgment transfer was conducted on the `MS MARCO` corpus. The `MS MARCO` crawl is a widely used training and test dataset in information retrieval, available both as a document corpus and as a passage corpus. Fröbe et al. [10] analyzed why retrieval models trained on `MS MARCO v1` performed better than those trained on `MS MARCO v2`. In version one, the passage dataset was available and judged before the document dataset. To take advantage of this, relevance judgments were directly transferred from the passage dataset to the document dataset by assigning the same relevance judgment to any document that had the same URL as the judged passage. A similar process was applied to enrich version two, where documents were assigned the same relevance judgment as their corresponding documents from version one if they had the same URL. The issue with this approach was that the document corpus in version one was crawled one year after the passage corpus, during which time the content of URLs remained largely unchanged. However, the document corpus in version two was crawled four years after the passage corpus, leading to many content changes. This dissimilarity resulted in inaccurate relevance transfers. This example highlights the importance of considering document content when transferring relevance judgments between datasets. To address this challenge, the transfer pipeline in this thesis employs a pairwise preference approach to compare old and new documents, ensuring a more reliable transfer of relevance information.

A more automated approach to infer relevance judgments was introduced by MacAvaney and Soldaini [18], who proposed One-Shot Labelers (1SL) to predict relevance for unjudged passages using nearest neighbor searches and various prompting strategies. Their goal was to examine the potential of large language models to fill „holes“ (i.e., unjudged documents) in relevance assessments for information retrieval evaluations. Their results demonstrated that instruction-tuned models, such as `FLAN-T5` [6], can provide reliable relevance estimations. However, their study focused only on passages and left document level inference as future work. This thesis addresses that gap by employing a pairwise inference approach with `FLAN-T5` to generate labels at document level.

Re-ranking methods play a crucial role in improving document ranking effectiveness. Traditional approaches include pointwise, pairwise, and listwise ranking, each with a trade-off between efficiency and effectiveness. Pradeep et al. [24] introduced a multi-stage ranking framework that combines document expansion, pointwise ranking, and pairwise re-ranking to enhance retrieval performance. Their findings showed that pairwise ranking models, such as `DuoT5`, significantly improve ranking quality, even in zero-shot scenarios. Inspired by this, the relevance transfer pipeline in this thesis leverages pairwise inference to determine the relevance of documents in the target corpus.

One of the key challenges in the relevance transfer pipeline is the selection of candidate documents from the target corpus for relevance transfer. The selection is a critical step, as it determines which documents should be considered for relevance inference. MacAvaney et al. [19] highlighted a major limitation in re-ranking pipelines. They depend on the recall of the initial candidate pool, meaning that documents not retrieved in this initial stage cannot be re-ranked later. To address this issue, they proposed a graph-based adaptive re-ranking approach, which expands the candidate pool beyond the initial retrieval set by leveraging the clustering hypothesis [16]. This hypothesis states that closely related documents are often relevant to the same queries. Following this idea, this thesis explores various nearest neighbor strategies for selecting documents for relevance transfer.

Processing pairwise preferences can be computationally expensive, particularly when dealing with large document collections, since it requires evaluating k^2-k preferences for k documents. To save computing resources, Gienapp et al. [15] investigated whether sampling from all possible preference pairs could improve the efficiency of pairwise re-ranking models without sacrificing effectiveness. Their findings showed that re-ranking effectiveness could be maintained with only one-third of the usual comparisons, with only a minor performance drop when further reducing comparisons. Based on that, the number of pairwise comparisons in the relevance transfer pipeline must not be exhaustive to maintain effective. Similarly, Zhuang et al. [32] introduced „Setwise prompting “, a novel zero-shot document ranking approach designed to reduce the number of required inferences while balancing efficiency and effectiveness. Their results demonstrated the advantages of Setwise prompting over traditional pairwise methods, suggesting that it could serve as a more fine-grained alternative for pairwise inference in relevance transfer.

Overall, these studies provide essential insights into relevance transfer, candidate retrieval, and efficiency optimizations, forming the foundation for the relevance transfer pipeline developed in this thesis.

Chapter 3

Methodology

In this Chapter, I present the individual steps of the developed pipeline used for relevance transfer. The process begins with an information retrieval task consisting of a document corpus, a set of queries, and corresponding relevance judgments, see Figure 3.2. The goal of the pipeline is to transfer the information of existing relevance judgments from a retrieval task to a target document corpus, thereby generating new relevance judgments. A simplified overview of the process is shown in Figure 3.1, which will be explained in this chapter.

The first step in the transfer pipeline involves selecting and segmenting documents from the retrieval task's document corpus. Only documents that contain at least one relevance judgment are selected, as those without any judgment do not provide useful information for the transfer process. Once selected, these documents are segmented into passages. This segmentation is done because relevant information related to an information need is typically concentrated in specific sections rather than spread across the entire document. By breaking documents into passages, the focus remains on the most relevant content. Additionally, since later stages of the pipeline rely on large language models, limiting the size of processed text through segmentation helps prevent potential issues when handling long documents with transformer models.

The second step focuses on identifying the most relevant passages within each selected document. Therefore, the relevance of each passage is determined in respect to the query of its document's relevance judgment. To achieve this, each passage is treated as an independent query and submitted to the source document corpus. The resulting document ranking is then used to compute various evaluation metrics, which are assigned as passage scores. These scores are later used for selecting candidate documents from the target document corpus and identifying source passages for pairwise preference comparisons.

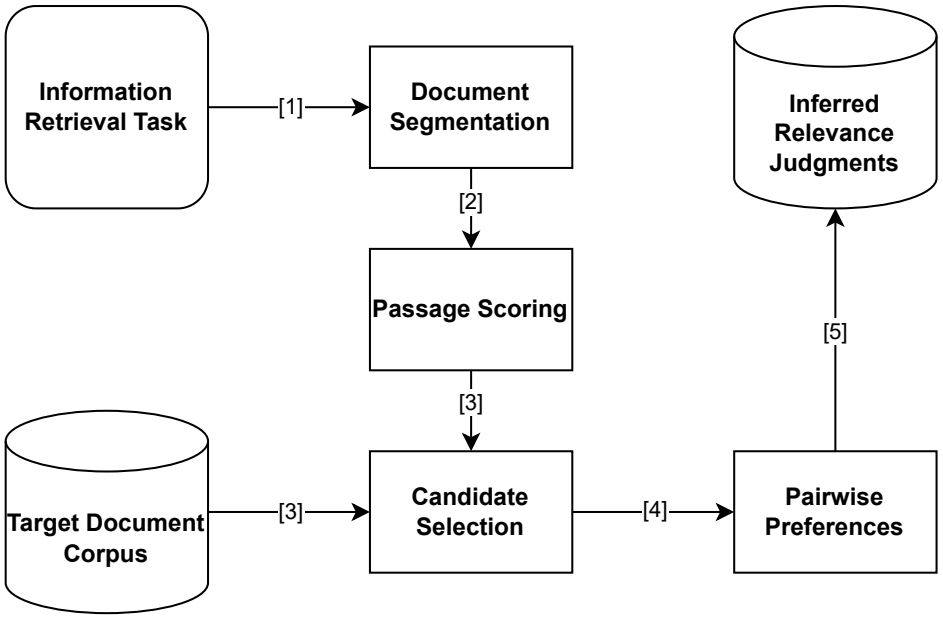


Figure 3.1: Overview of the transfer pipeline and its individual steps to transfer exisiting relevance judgments into another document corpus.

The third step is the selection of candidate documents from the target document corpus. For each query in a retrieval task, a set of documents is chosen to receive new relevance judgments. This selection is performed using different strategies with the aim to identify documents most likely to be relevant. Each selected target document is then also segmented into passages and each passage is paired with relevant passages from the source dataset, identified in the previous passage scoring step. These candidates serve as input for the pairwise inference process in the last step.

At this stage, the transfer pipeline has identified, for each query, a set of target documents along with a corresponding set of relevant passages from the source dataset. The final step is inferring new relevance judgments for the target documents. Therefore, each query is processed alongside a target passage and a source passage using a pairwise ranking model to determine whether the target passage is as relevant for the query as the source passage. Finally, a judgment for the entire target document is created by aggregating all of its inferred passage scores.

Table 3.1: Source corpora and their retrieval tasks from which query relevance judgments (`qrels`) were transferred to the target document corpus `ClueWeb22/b`.

Corpus		Associated Retrieval Task			
Name	documents	Name	Queries	`qrels`	Labels
Args.me [1]	0.4 m	Touché 2020 Task 1 [4]	49	2 298	3
Disks4+5 [29]	0.5 m	Robust04 [28]	250	311 410	3
		TREC-7 [30]	50	80 345	2
		TREC-8 [31]	50	86 830	2
MS MARCO Passage [3]	8.8 m	TREC 2019 DL Track [9]	43	9 260	4
		TREC 2020 DL Track [8]	54	11 386	4

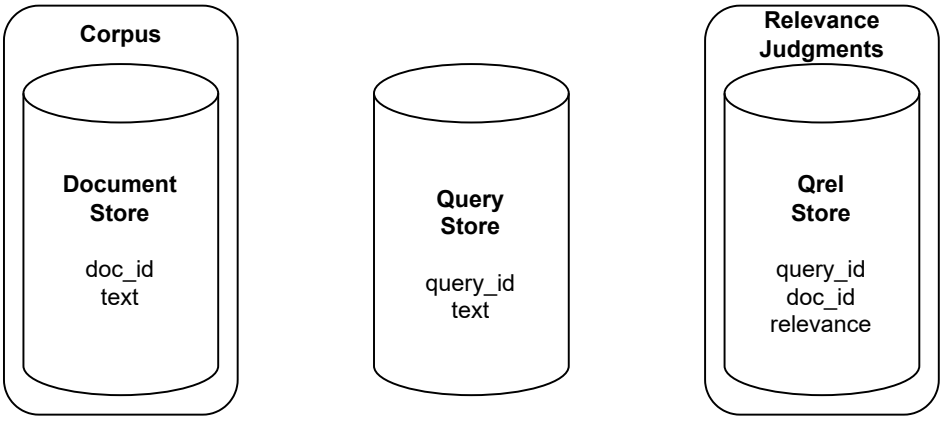


Figure 3.2: Simplified structure of a dataset from `ir_datasets`, comprising a set of documents, queries, and relevance judgments, along with their attributes.

3.1 Datasets

Before the actual processing can begin, it is necessary to determine the source and target of the relevance transfer. This involves selecting datasets, along with their associated information retrieval tasks, as the foundation for transferring relevance judgments to a target document corpus.

To simplify data handling, `ir_datasets` [20] was used, a Python package that provides access to numerous information retrieval datasets. As illustrated in Figure 3.2, each dataset consists of a document corpus, a query store, and a set of query relevance judgments for the queries and documents.

A key advantage of `ir_datasets` is its standardized interface[1], which enables uniform access to different datasets. Through built-in iterators, the package facilitates structured access to corpora, queries, and relevance judgments, allowing the transfer pipeline to handle diverse datasets efficiently.

The corpora and associated information retrieval tasks used in this thesis are listed in Table 3.1. The source datasets were selected based on their widespread use in information retrieval research and their varying sizes. `Args.me` and `Disks4+5` are relatively small corpora, whereas `MS MARCO Passage` is significantly larger. Additionally, the number of existing relevance judgments in these datasets varies considerably. `Args.me` contains approximately 2 300 judgments, while the retrieval tasks associated with `MS MARCO` have several thousand. In contrast, the tasks associated with `Disks4+5` extend far beyond this, with one exceeding 300 000 judgments. Notably, the `MS MARCO` dataset is already preprocessed and provided as passages, unlike the other datasets. This is a significant difference, as the relevance judgments are already at passage level, eliminating the need for further segmentation of the documents. This diversity in dataset size and relevance judgment density provides a robust test for the transfer process under different conditions.

The selected datasets serve as the starting point of the pipeline, while the final target corpus for relevance transfer is `ClueWeb22/b`. This dataset was chosen because it is the newest ClueWeb corpus in the `Lemur Project`[2]. With over 1.0 billion documents, `ClueWeb22/b` is significantly large, but due to its recent release, it currently has a low number of relevance judgments. The objective of the transfer process is to enrich `ClueWeb22/b` by leveraging the existing relevance judgments from the source datasets, thereby enhancing its use cases for information retrieval research.

[1] https://ir-datasets.com/python.html
[2] https://lemurproject.org

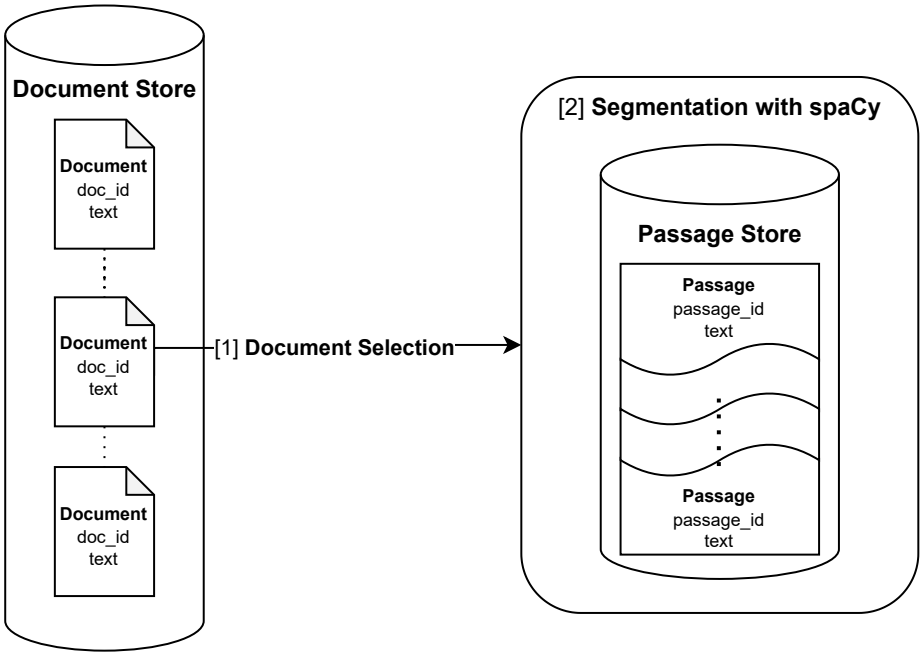


Figure 3.3: Visualization of the document segmentation step in the transfer pipeline. First, a subset of documents is selected from the source document store. Then, the selected documents are segmented using `spaCy`.

3.2 Document Segmentation

With the selection of the source datasets and the target document corpus completed, the actual processing of the transfer pipeline can begin. To provide a clearer understanding of the individual steps involved, the following explanations will use one source dataset from Table 3.1 as an illustrative example. The described steps apply uniformly across all corpora and retrieval tasks.

The first step involves selecting and segmenting documents from the source dataset into passages, as illustrated in Figure 3.3. The primary objective of the pipeline is to transfer the relevance judgments from the source retrieval tasks to the target corpus. This transfer starts by identifying the relevant documents in the source corpus based on the existing relevance judgments for each query in the retrieval task. These documents serve as the foundation for the later pairwise preference step, where they are compared with documents from the target corpus. To reduce computational overhead, only a subset of documents from the source dataset is selected. As demonstrated by Gienapp et al. [15],

the number of pairwise preferences can be reduced through sampling without significantly impacting effectiveness. Consequently, the pipeline will utilize a limited set of comparison documents rather than all possible documents, minimizing the computational cost. Once the relevant documents have been selected, they are segmented into passages. This segmentation is essential because relevant information related to a query is often confined to specific sections rather than the entire document. By processing documents at the passage level, the pipeline can focus on the most relevant parts while filtering out less relevant parts of a document. Additionally, the subsequent steps of the pipeline involve processing text with transformer models, which have a maximum input length they can handle. Segmenting documents into smaller passages ensures compliance with these constraint, thereby enabling the models to operate effectively [17].

3.2.1 Document Selection

Instead of using the entire document corpus for relevance transfer, only a representative selection of documents is used, step one of Figure 3.3. First, all documents without relevance judgments in the retrieval task can be ignored since they contain no useful information for the relevance transfer. Therefore, only judged documents are considered for selection. A document is considered judged if it has at least one relevance judgment in the `qrel store`. These judged documents, along with their relevance judgments for one or multiple queries of the retrieval task, can be used to transfer information, as their relevance has already been determined.

As shown in Table 3.1, each retrieval task uses different relevance labels to indicate the relevance of a document to a given query. Some datasets apply binary labels, distinguishing simply between relevant and non-relevant documents, while others utilize a Likert scale with multiple relevance levels to represent varying degrees of relevance. In this thesis, no distinction is made between different levels of non-relevance. For instance, „not relevant“ (`label = 0`) and „strongly not relevant“ (`label < 0`) are treated equally. This standardization was applied because finer distinctions between non-relevant documents are not considered necessary for the pipeline, as many TREC evaluation implementations and metrics, such as `nDCG`, ignore negative labels by mapping them to zero by default [14].

At this stage, all documents with at least one relevance judgment for any query of the retrieval task have been identified. Some retrieval tasks, such as `Robust04` with over 300,000 relevance judgments, contain a large number of judgments per query. As mentioned above, only a representative subset of judged documents can be used for relevance transfer for each query. To manage this, the number of judged documents in the transfer process is limited. For each query, a maximum of 50 relevance judgments per relevance label of the retrieval task, referred to as a query-label combination, is selected.

3.2.2 Segmentation with spaCy

Now that the documents used for relevance transfer have been selected, they are segmented into passages. This segmentation is performed to focus only on the most relevant parts of a document rather than the entire text and to ensure proper processing of text snippets through large language models later. To achieve accurate segmentation with correct sentence separation, the GitHub repository `grill-lab/trec-cast-tools`[3] was utilized. The repository provides a collection of scripts designed to process the TREC CAsT Track 2022 [22]. Among its features is the ability to process a document collection and generate passage splits.

`trec-cast-tools` leverages `spaCy`[4], a powerful natural language processing library in Python. `spaCy` offers various features such as: tokenization, part-of-speech tagging, named entity recognition, lemmatization, and many more. It also provides functionality for sentence segmentation, which is used by `trec-cast-tools`. First, the documents are processed by `spaCy`, splitting the texts into sentences. Then, `trec-cast-tools` concatenates these sentences into passages, with each passage limited to a maximum length of 250 words. After processing all the selected documents from the source corpus, they are now stored as uniquely identifiable passages, ensuring traceability throughout the transfer pipeline. These passages are saved in the `passage store`, as shown in the second step of Figure 3.3.

When a document is judged relevant to a query, the relevant information is often scattered throughout the text. Therefore, the next step in the transfer pipeline is to identify the most relevant passages within each document. This step optimizes the subsequent candidate selection process and improves the effectiveness of the pairwise preference evaluation by ensuring that highly relevant passages from the source corpus are used for comparison.

[3] https://github.com/grill-lab/trec-cast-tools
[4] https://spacy.io

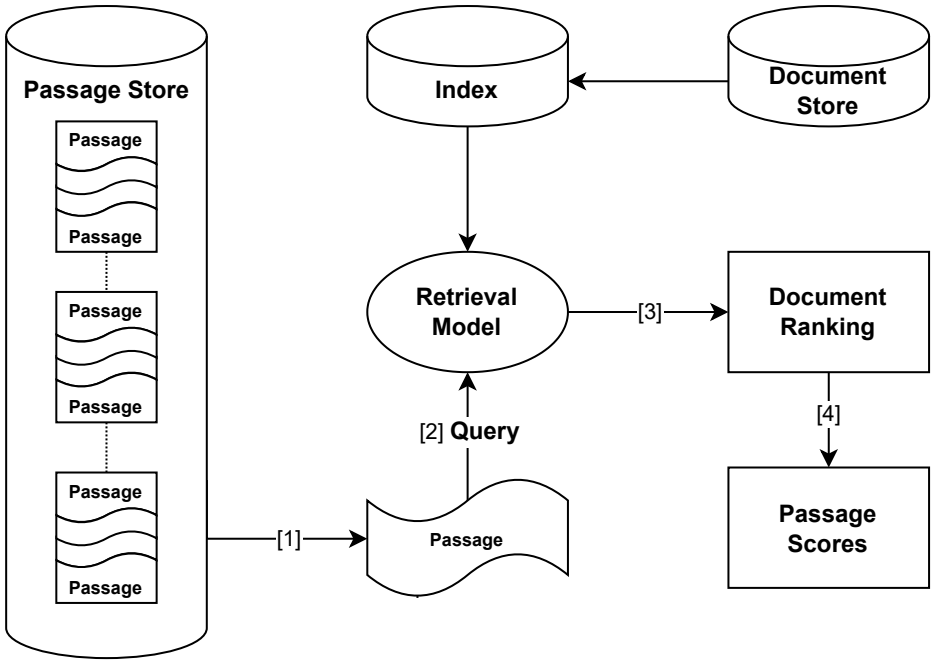


Figure 3.4: Overview of the passage scoring process, where each passage is submitted as an independent query to its source document corpus, and scores are assigned based on the retrieved document ranking.

3.3 Passage Scoring

As selected in Section 3.2.1, each query-label combination of a retrieval task has an associated set of relevance judgments along with their corresponding documents. To determine which passages of a document are most relevant and best reflect the label of its relevance judgment, a ranking of the individual passages is performed in this step. To rank all passages of the selected documents, the following procedure is applied to all passages. First, each passage of a document is treated as an independent query, as shown in step one and two of Figure 3.4. This query is then used to retrieve a document ranking from its original source `document store`, as shown in step three of Figure 3.4. Based on the retrieved document ranking for the submitted passage, the relevance of the passage is determined by computing its `precision@10` and `nDCG@10` scores, step four of Figure 3.4. These metrics were chosen because they are widely used in information retrieval research. In contrast to `precision`, `nDCG` provides a more fine-grained evaluation by considering relevance labels of the retrieved documents, whereas `precision` only accounts for binary relevance.

Precision is the fraction of retrieved documents that are relevant to an information need. In this context, the information need corresponds to the original query associated with the relevance judgment of the document to which a passage belongs. A retrieved document is considered relevant if it has been judged as such in the relevance judgments for that query. To simplify the scoring process, the evaluation is restricted to the top 10 retrieved documents. The resulting `precision@10` score is then assigned as the passage's score, representing its relevance to the query of the document's original relevance judgment.

Normalized Discounted Cumulative Gain (`nDCG`) is another metric commonly used in information retrieval to evaluate the quality of a retrieved document ranking. Similar to `precision`, the query associated with the relevance judgment of the document to which a passage belongs is used to determine the relevance labels of the retrieved documents. Cumulative Gain (`CG`) represents the sum of all relevance labels for the retrieved document ranking. Unlike `precision`, `CG` takes into account the label values rather than simply differentiating between relevant and non-relevant labels. This provides greater granularity in retrieval tasks with more than two relevance labels, allowing for more nuanced scoring of document passages. The advanced Discounted Cumulative Gain (`DCG`) further refines this evaluation by introducing a positional factor to the ranking, assigning higher weight to relevant results that appear earlier in the ranking. The final `nDCG` score is calculated by normalizing the `DCG` score with the Ideal DCG (`IDCG`), which represents the optimal theoretical document ranking for the query. As with `precision`, the evaluation is limited to the first 10 documents by only computing `nDCG@10`.

The idea behind ranking a document's passages is that passages containing a high density of relevant information to an information need (i.e., the original query of a relevance judgment) are more likely to retrieve relevant documents and therefore achieve higher `precision@10` and `nDCG@10` scores. Conversely, passages with minimal or no relevant information will lead to lower scores due to fewer retrieved relevant documents.

At this stage of the pipeline, the individual passages of the selected documents have been scored. These scores will serve as the foundation for the subsequent steps in identifying candidate documents from the target corpus. The selected candidates will receive newly inferred relevance judgments at the end of the pipeline. Additionally, these scores will help determine the most relevant passages from the source document corpus, which will be used in pairwise preference comparisons alongside the selected target candidates.

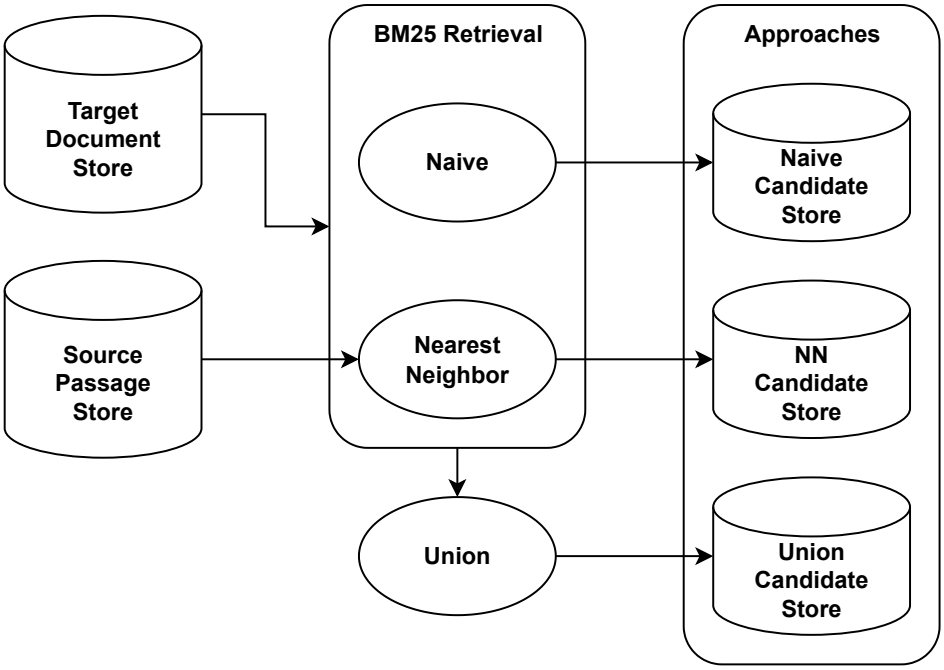


Figure 3.5: Overview of the three candidate retrieval approaches for selecting documents from the target corpus. The selected candidates will receive new relevance labels inferred at the end of the pipeline.

3.4 Candidate Selection

The next step in the transfer pipeline is selecting candidates for pairwise preference inference. This step consists of two parts. First, for each query in a retrieval task, documents from the target corpus are selected. New relevance judgments will be created for these queries and their selected documents. The second part involves choosing passages from the source corpus for each query. These passages will be used in pairwise preference comparison to determine whether a document from the target corpus is relevant to a given query.

3.4.1 Document Retrieval from Target Dataset

For each query in the retrieval task of the source dataset, a set of documents from the target corpus must be selected for which new relevance judgments will be inferred. The goal is to identify documents that are most likely relevant to a query. By pre-selecting potentially relevant documents, the number of identified relevant documents at the end of the pipeline should be increased.

To achieve this, all three tested approaches for selecting target documents begin with `BM25` retrieval as a pre-selection method. By submitting texts or passages, that are already identified as relevant for a query, against the target document store, a ranking of target documents is retrieved. The highest-ranked documents are most likely to be relevant to the query and therefore qualify for relevance inference. A fine-grained classification of relevance is performed later through the actual pairwise preference inference. `BM25` retrieval was chosen for its fast retrieval and because only an initial rough assessment is required.

Naive The first approach, termed `naive`, submits each original query text from a retrieval task to the target corpus to retrieve potentially relevant documents. From the resulting document ranking, the top 1000 documents are selected as candidates. Unfortunately, some queries may be ambiguous, leading to multiple possible interpretations. For example, the query „Apple“ could refer to either the technology company or the fruit. To address such ambiguities, the query narrative is also submitted as an independent query. A query narrative is a brief text that provides additional context, clarifying the search intent and specifying a query's focus. This retrieves another 1000 documents via `BM25` ranking. After filtering out duplicates between the two sets, up to 2000 unique documents per query are selected.

Nearest Neighbor The second approach, `nearest neighbor`, is based on the relevant passages identified in Section 3.3. In this strategy, the top-scoring passages for each query are submitted as queries to the target corpus. For each passage, the top 20 documents in the retrieved document ranking are selected as candidates for the corresponding query. As a result, each passage can contribute up to 20 unique documents to the candidate set. Additionally, several variations of the `nearest neighbor` approach were tested to determine the most effective candidate selection strategy. First, the number of top passages per query from the source dataset was limited. Three variations were evaluated: one using the top 10 passages, another using the top 50, and a third using the top 100 passages. This limitation significantly reduces the number of potential candidates, as a single query can have an exceedingly large number of relevant passages. The second variation is restricting the selection to one passage per source document. For example, if the top 10 passages for a query contained multiple passages from the same document, only the best-scoring passage is used for retrieving. The intention behind this restriction is to ensure greater diversity in the retrieved candidates. Conversely, an alternative approach permits multiple passages from the same document. This variation was tested to assess the impact of passage diversity on retrieval effectiveness.

Union Approach The third approach, called `union`, combines the `naive` and `nearest neighbor` approaches into a single candidate set for each query. This combination is intended to enhance the recall of retrieved relevant documents by including a broader set of potentially relevant documents. However, the increased number of documents will reduce precision by allowing more non-relevant documents to be included, thereby increasing the workload for pairwise preference evaluations in later stages. A detailed evaluation of the various candidate retrieval approaches is done in Section 4.4.

3.4.2 Postprocessing of Selected Target Documents

As described in Section 3.2.2, selected documents from the target corpus are segmented into passages for later processing. Since pairwise preferences will be applied at passage level, documents from both the source and target datasets have to be divided into passages. Therefore, the selected candidate documents are processed with `trec-cast-tools` as outlined before. First, `spaCy` is employed for segmenting the chosen target documents into sentences, and then passages are formed by concatenating these sentences.

3.4.3 Composing Final Candidates

For pairwise preference inference, a query, a passage from a selected document in the target corpus, and a known passage from the source corpus are required, as illustrated in Figure 3.6. The queries are provided directly by the retrieval task and, as described above, the target documents have been identified and segmented into passages. What remains is the selection of passages from the source corpus for pairwise preference comparison. Therefore, the goal is to compare each selected passage from the target corpus, for a given query, against 15 relevant and 5 non-relevant passages from the source corpus.

Simple Selection During the passage scoring stage of Section 3.3, all passages of each selected document from the source corpus are evaluated and assigned scores based on `precision@10` and `nDCG@10` metrics. These scores are now used to rank all passages for each query. Following this ranking, the 15 highest-scoring relevant passages and the 5 lowest-scoring non-relevant passages for each query are selected for pairwise preference comparison.

Diversified Selection It is possible that the top- and lowest-rated passages are predominantly drawn from a small subset of documents. This can occur when multiple passages from the same document receive extremely high or low passage scores. While such passages may meet the selection criteria of the `simple selection` approach, this concentration could unintentionally reduce the diversity of pairwise preference comparisons. To avoid potential bias resulting from limited document diversity, this approach restricts the selection to a maximum of one passage per document. Specifically, the approach selects the top 15 and bottom 5 passages for each query, ensuring that each passage is from a distinct document. This strategy enhances the diversity of pairwise preference comparisons by forcing a broader range of source documents.

The selected 20 passages from the source corpus using either the `simple selection` or `diversified selection` approach are now paired with the selected passages from the target corpus. These pairs serve as the input for pairwise preference inference to determine the relevance of the target passages.

3.5 Pairwise Preferences

So far, the transfer pipeline has selected and segmented a subset of already judged documents from the source corpus for each query. Each passage was then scored to estimate its relevance for later processing steps. Subsequently, various approaches for selecting candidates for each query of a retrieval task for pairwise preference inference were introduced. As the output of Section 3.4, each selected candidate document from the target corpus has been segmented into passages and each of these passages is paired with the top 15 relevant and bottom 5 non-relevant passages from the source corpus. The queries used for the pairwise preferences are predefined by the corresponding retrieval task. At this point, all necessary preprocessing steps have been completed, and the candidates, consisting of `(query, known passage, passage to judge)`, are now ready for pairwise preference processing, as illustrated in Figure 3.6.

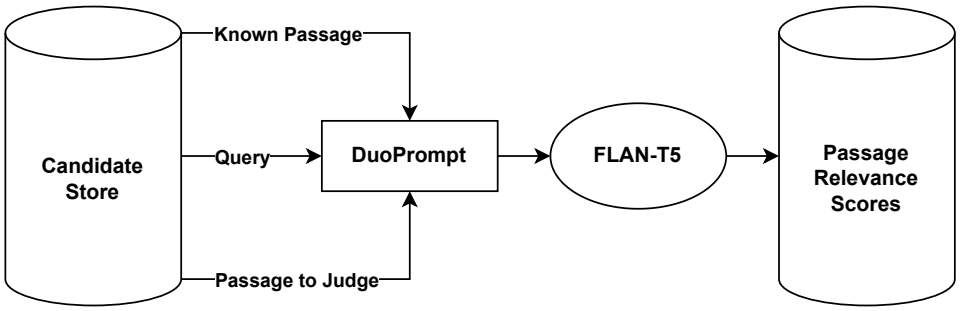


Figure 3.6: Visualization of the pairwise preference inference process with `DuoPrompt`. Each candidate consists of a query, a known passage from the source corpus, and a passage to judge from the target corpus.

DuoPrompt

```
PROMPT = (
    "Determine if passage B is as relevant as passage A "
    "for the given query. "
    'Passage A: "...{{ rel_doc_text | replace("\\"", "\'") }}..." '
    'Passage B: "...{{ unk_doc_text | replace("\\"", "\'") }}..." '
    'Query: "{{ query_text }}" '
    "Is passage B as relevant as passage A? </s>"
)
```

Figure 3.7: One-shot `DuoPrompt` prompt instructing the model in Figure 3.6 to assess whether the passage to judge is as relevant as the known passage for a query.

To perform the pairwise preference inference, the Python library `autoqrels`[5] was used. `autoqrels` is a tool designed for automatically inferring relevance judgments. It supports zero-shot and one-shot prompting for relevance inference and supports several pre-implemented inference models. The original `autoqrels` paper [18] evaluated the effectiveness of one-shot labelers for automatic relevance estimation. Among the tested approaches `MaxRep-BM25`, `MaxRep-TCT`, `DuoT5`, and `DuoPrompt`, the `DuoPrompt` approach for pairwise inference demonstrated superior performance and therefore is utilized for one-shot labeling in this thesis.

[5] https://github.com/seanmacavaney/autoqrels

Figure 3.7 illustrates the `DuoPrompt` structure used for pairwise inference with `FLAN-T5`. The prompt takes a query (`query_text`), a known passage (`rel_doc_text`), and a passage to judge (`unk_doc_text`) as inputs. It includes a brief instruction for the model, asking it to compare the passage to judge with the known passage and assess its relevance to the query. The model then predicts a relevance score for the passage to judge. This structured comparison helps the model determine whether the passage to judge is as relevant as the known passage. If it is highly relevant, the expected score should be close to 1.0. Conversely, if it is not relevant at all, the score should be close to 0.0.

`FLAN-T5` [6] is a Text-to-Text Transfer Transformer model, an advanced version of Google's `T5` model [25]. While maintaining the backbone architecture of `T5`, `FLAN-T5` has been further fine-tuned on a diverse set of training tasks, enhancing its performance across various natural language processing applications. `FLAN-T5` is available in multiple sizes, ranging from smaller, resource-efficient versions like `flan-t5-small`, with 80 million parameters, to models such as `flan-t5-xxl`, which contains 11 billion parameters. These variations accommodate different computational and application needs. Given the computational constraints for this thesis, `flan-t5-base`[6] model, with 250 million parameters, is selected as the transformer model for `DuoPrompt`.

All candidates undergo pairwise preference inference, where each passage to judged is compared against 20 known passages, as outlined in Section 3.4.3. As a result, the pairwise preference inference produces 20 relevance scores for each passage, with higher scores indicating greater relevance to the query. The overall relevance of each passage is then computed by aggregating its inferred scores, a process that will be detailed in the evaluation of Section 4.5. To determine the final relevance score for a document, the maximum of all its passage scores is used. This approach follows the principle that a document's relevance is determined by its most relevant passage [9].

This chapter provided a comprehensive description of all steps in the transfer pipeline, resulting in a set of relevance scores for each passage to judge. These scores represent the transferred information of the relevance judgments from a source dataset's retrieval task into a target document corpus. The next chapter will evaluate the effectiveness of the individual steps and the overall quality of the transferred relevance judgments across datasets.

[6] https://huggingface.co/google/flan-t5-base

Chapter 4

Evaluation

This chapter evaluates the developed relevance transfer pipeline. First, the individual steps of the pipeline are analyzed to examine the intermediate results and identify potential weaknesses, including document selection, passage scoring, and candidate selection. Document selection and passage scoring are performed solely on the source datasets and do not depend on the target dataset. However, to assess the quality of the candidate selection approaches, the transfer pipeline is applied to the same dataset as both the starting and target point of the relevance transfer. Evaluating candidate retrieval requires a target dataset with existing relevance judgments for the queries of the source dataset. Since candidate retrieval aims to pre-select documents from a target corpus that are likely relevant to a query, these judgments are necessary to determine how many selected documents are truly relevant.

The second part of the chapter evaluates the inferred relevance judgments through two transfer strategies. The first evaluation applies the transfer pipeline to a source dataset and uses the same dataset as the target. This self-transfer evaluation measures how well the pipeline reproduces known relevance judgments within the same dataset. The second evaluation assesses the final transfer to `ClueWeb22/b` as the target dataset. Since `ClueWeb22/b` does not provide relevance judgments for the tested source retrieval tasks, a pooling process is conducted on the candidate documents retrieved by a candidate selection approach. This is followed by a manual relevance judgment of the pooled documents. The quality of the transferred relevance judgments to `ClueWeb22/b` is then evaluated based on these manual judgments.

Table 4.1: Comparison of rank correlation metrics computed by Kendall's τ and Spearman's ρ. The table shows the rank correlation between the reference scores $[0.2, 0.7, 0.5]$ and three label sets: two with an expected correlation of 1, and one with lower correlation. `Default` represents the rank correlation scores computed by the standard algorithms, while `greedy` shows the scores obtained by greedily mapping the reference scores to the label set, representing the idealized rank correlation outcome.

Comparative Set	**Default**		**Greedy**	
	τ	ρ	τ	ρ
$[0, 2, 1]$	1.00	0.99	1.00	1.00
$[0, 1, 1]$	0.82	0.92	1.00	1.00
$[0, 0, 1]$	0.00	0.11	0.50	0.50

4.1 Rank Correlation

Before starting with the evaluation of the transfer pipeline, I present the metrics used to evaluate the relevance judgments produced by the pipeline against the actual relevance judgments provided by the retrieval tasks. This clarification is important, as the evaluation of the assigned passage scores and the final relevance judgments generated by the pipeline are based on these metrics.

The `Inter-Annotator Agreement` [2] is a statistical measure that quantifies the consistency among multiple annotators when labeling the same dataset. It provides insight into annotation quality by indicating the level of agreement or disagreement over all annotations. However, this measure requires all annotators to use the same categorical label set. In this thesis, retrieval tasks assign integer relevance labels ($\in \mathbb{N}$), while the inferred relevance scores are continuous scores ($\in \mathbb{R}^+$). Due to this mismatch, the `Inter-Annotator Agreement` is unsuitable for evaluation.

Instead, `Rank Correlation` is used to compare the actual relevance labels with the inferred relevance scores. `Rank Correlation` evaluates how well two rankings align. Here, the retrieval task's relevance judgments serve as the ground truth, against which the transfer pipeline's inferred judgments are compared. The rank correlation metrics used are `Kendall's` τ and `Spearman's` ρ. Both metrics compute the rank correlation between linear orders [21], making them well suited for the retrieval task labels and the pipeline's inferred scores. A high rank correlation indicates that the generated judgments preserve the ranking order of the original relevance labels, which is the goal.

Table 4.1 illustrates how these metrics assess some sample relevance judgments. It presents correlation between example reference scores $[0.2, 0.7, 0.5]$ and three label sets: the first two yield an expected correlation of 1, while the third is expected to have a lower correlation. The `default` column contains correlation scores computed by standard `Kendall's` τ and `Spearman's` ρ algorithms. Unlike the default algorithms, the goal in this thesis is to maximize alignment between inferred relevance scores and ground truth labels. For instance, the relevance scores $[0.2, 0.7, 0.5]$ should ideally correspond to the label set $[0, 1, 1]$ in the second row of Table 4.1. This alignment can be achieved by mapping scores less than or equal to 0.2 to label 0 and scores greater than 0.2 to label 1. Therefore, the intended rank correlation for this example should be 1.

To accomplish this, a modified version called `greedy` was tested alongside the default algorithms. This variant greedily maps the relevance scores to the label set, producing the highest possible rank correlation. The example demonstrates that the standard correlation metrics are sensitive to the rank order of relevance judgments, whereas the `greedy` version eliminates this sensitivity, achieving the maximum correlation scores.

4.2 Document Selection

The first step in the transfer pipeline was selecting and segmenting a subset of documents for each query in a retrieval task from the source corpus. These selected documents play a crucial role in two subsequent steps, identifying candidate documents from the target corpus and performing pairwise preference comparisons using `DouPrompt`. Since only a subset of documents is required per query, two selection criteria were applied. First, only documents with at least one relevance judgment in the `qrel store` were considered, as documents without judgments do not provide transferable information. Second, a maximum of 50 relevance judgments per relevance label per query, referred to as a query-label combination, was selected to manage computational complexity.

Table 4.2 presents the results of the selection process. Based on the selection criteria, a maximum of 50 documents per query-label combination is possible. Ideally, each query should have around 50 non-relevant documents and at least 50 relevant ones. All `TREC` retrieval tasks performed well, reaching the maximum number of non-relevant documents per query. `TREC-7` and `TREC-8` also showed strong results with an average over 40 relevant documents per query. `Robust04`, despite having a large number of relevance judgments for its 250 queries, does not reach an average of 50 relevant documents per query.

Table 4.2: The table presents the results of the candidate selection process, showing for each label of a retrieval task the number of documents per query for each source dataset. A maximum of 50 documents per query-label combination is possible.

Docs./Query	Touché 20	Robust04	TREC-7	TREC-8	TREC-19 DL	TREC-20 DL
Label 0	27.9	50.0	50.0	50.0	49.6	50.0
Label 1	6.0	32.8	40.2	40.5	31.1	28.5
Label 2	13.0	3.8	-	-	23.8	16.0
Label 3	-	-	-	-	11.7	10.3
Total	46.9	86.6	90.2	90.5	116.2	104.8

This is because only 5.6% [29] of its judgments are relevant judgments. Nonetheless, it maintains a solid average of over 36 relevant documents per query for further processing. Additionally, `TREC-19 DL` and `TREC-20 DL` exceeded the threshold of 50 relevant documents per query, ensuring a diverse pool for candidate retrieval and pairwise comparisons. The only outlier is `Touché 20`, which achieves an average of just 28 non-relevant and 19 relevant judgments per query. The low number of relevance judgments in this retrieval task could limit the effectiveness of the nearest neighbor candidate retrieval approach and the pairwise preference comparisons. However, given the relatively small size of `Args.me` with 0.4 million documents, the pipeline's results may still be valuable.

This intermediate evaluation confirms that the selection process successfully provides an adequate number of documents for subsequent steps in the pipeline. While `Touché 20` remains useful, its results should be interpreted cautiously due to the lower number of judged documents per query.

Table 4.3: Average rank correlations between assigned passage scores and original relevance judgments across all queries of a retrieval task, reported using standard Kendall's τ and Spearman's ρ.

	Retrieval Model	Touché 20		Robust04		TREC-7		TREC-8		TREC-19 DL		TREC-20 DL		Avg.	
		τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ
nDCG@10	BM25	0.802	0.871	0.823	0.907	0.801	0.896	0.813	0.906	0.725	0.831	0.755	0.846	0.787	0.876
	DFR_BM25	0.804	0.874	0.822	0.907	0.800	0.896	0.813	0.906	0.727	0.833	0.755	0.846	0.787	0.877
	DFIZ	0.778	0.852	0.820	0.906	0.796	0.896	0.800	0.898	0.730	0.836	0.753	0.845	0.780	0.872
	DLH	0.839	0.901	0.831	0.913	0.810	0.903	0.825	0.915	0.735	0.839	0.754	0.845	0.799	0.886
	DPH	0.783	0.856	0.820	0.905	0.799	0.895	0.809	0.903	0.725	0.831	0.753	0.844	0.782	0.872
	DirichletLM	0.679	0.752	0.816	0.902	0.798	0.896	0.802	0.898	0.721	0.825	0.751	0.841	0.761	0.852
	Hiemstra_LM	0.855	0.914	0.829	0.912	0.807	0.902	0.819	0.910	0.729	0.834	0.759	0.849	**0.800**	**0.887**
	LGD	0.801	0.874	0.819	0.905	0.799	0.897	0.802	0.899	0.730	0.835	0.757	0.848	0.785	0.876
	PL2	0.805	0.873	0.826	0.909	0.804	0.898	0.822	0.913	0.727	0.833	0.757	0.848	0.790	0.879
	TF_IDF	0.807	0.876	0.823	0.907	0.801	0.896	0.814	0.907	0.726	0.832	0.755	0.847	0.788	0.877
precision@10	BM25	0.765	0.828	0.807	0.881	0.792	0.872	0.807	0.885	0.657	0.756	0.697	0.785	0.754	0.835
	DFR_BM25	0.768	0.833	0.806	0.880	0.791	0.870	0.807	0.884	0.658	0.757	0.696	0.785	0.754	0.835
	DFIZ	0.729	0.798	0.801	0.874	0.785	0.863	0.791	0.869	0.661	0.760	0.695	0.784	0.744	0.824
	DLH	0.802	0.862	0.817	0.889	0.803	0.881	0.823	0.899	0.666	0.764	0.697	0.785	**0.768**	**0.847**
	DPH	0.745	0.814	0.803	0.876	0.787	0.867	0.801	0.878	0.659	0.758	0.697	0.785	0.749	0.830
	DirichletLM	0.640	0.705	0.796	0.868	0.785	0.863	0.791	0.867	0.652	0.750	0.691	0.778	0.726	0.805
	Hiemstra_LM	0.819	0.880	0.813	0.886	0.799	0.878	0.815	0.892	0.658	0.757	0.699	0.786	0.767	0.846
	LGD	0.765	0.833	0.800	0.873	0.787	0.866	0.792	0.870	0.659	0.758	0.697	0.785	0.750	0.831
	PL2	0.766	0.829	0.811	0.885	0.796	0.876	0.819	0.896	0.656	0.755	0.698	0.786	0.758	0.838
	TF_IDF	0.769	0.832	0.808	0.881	0.793	0.872	0.808	0.885	0.657	0.756	0.697	0.785	0.755	0.835

4.3 Passage Scoring

The next step in the transfer pipeline was to assign relevance scores to the passages of the selected documents from the previous step. This was done to identify the most relevant passages of a document with respect to the actual query associated with a document's relevance judgment. To achieve this, each passage was treated as an independent query and used to retrieve a document ranking from its original source corpus. Based on the retrieved document rankings, `precision@10` and `nDCG@10` were computed and assigned as passage scores. Since this thesis does not aim to optimize or analyze specific retriever systems but rather utilizes them for passage scoring, a diverse selection of models was tested, as listed in Table 4.3.

To evaluate the quality of the assigned passage scores and determine which retrieval model and metric combination are most effective in determining the relevance of passages, the rank correlation between the original relevance judgments and the newly assigned scores was computed using Kendall's τ and Spearman's ρ. Therefore, for each retriever-metric combination, an overall document score was aggregated from its passage scores by assigning a document the maximum passage score of its passages [9].

The rank correlation was then computed for all judged documents of each query individually, rather than across all relevance judgments of a retrieval task simultaneously, and then averaged. Macro-averaging across all queries is intended to reduce the effect of potential outliers for individual queries.

The results in Table 4.3 show that all retrieval models achieved a very high rank correlation across all tested information retrieval tasks. While `Hiemstra_LM` was the best-performing model, all models demonstrated strong performance based on the average τ and ρ values. Therefore, the choice of retrieval model is not a critical factor for subsequent steps in the transfer pipeline, as all models produced passage scores with high rank correlation to the actual relevance labels from the retrieval tasks. This finding also applies to the `greedy` variant of both rank correlation methods, as shown in Table A.1. Although the `greedy` variant resulted in slightly higher rank correlation scores than the default version, the overall behavior remained consistent.

However, the choice of metric had a significant impact. `precision@10` consistently resulted in lower rank correlation compared to `nDCG@10`. This outcome was expected, as `precision@10` lacks the granularity needed to effectively differentiate between individual passages. Consequently, `nDCG@10` is the better metric for passage scoring. Therefore, in the following steps of the transfer pipeline, the passage scores assigned by `nDCG@10` alongside the retrieval model that achieved the highest rank correlation for each retrieval task is used.

Table 4.4: Overview of recall (Rec.) and the average number of candidate documents per query (Docs.) for all three candidate retrieval approaches, tested with document selection restricted to one passage per document (opd.) and without restriction (def.). The `Nearest Neighbor` and the `Union` approaches were also evaluated with varying k, determining the number of passages used for candidate retrieval.

Approach			Touché 20		Robust04		TREC-7		TREC-8		TREC-19 DL		TREC-20 DL	
		k	Rec.	Docs.	Rec.	Docs.	Rec.	Docs.	Rec.	Docs.	Rec.	Docs.	Rec.	Docs.
Naive	-	-	0.912	1 556	0.777	1 572	0.657	1 600	0.713	1 566	0.736	976	0.751	955
Nearest Neighbor	def.	10	0.609	97	0.530	88	0.430	85	0.449	89	0.560	81	0.616	83
		50	0.828	318	0.744	334	0.696	349	0.665	356	0.872	268	0.906	269
		100	0.978	479	0.833	618	0.787	657	0.757	684	0.966	436	0.974	407
	opd.	10	0.699	104	0.584	96	0.495	99	0.489	101	0.560	81	0.616	83
		50	**1.0**	209	0.876	360	0.826	419	0.797	439	0.872	268	0.906	269
		100	**1.0**	209	0.885	375	0.826	419	0.797	439	0.966	436	0.974	407
Union	def.	10	0.931	1 577	0.855	1 618	0.771	1 651	0.795	1 611	0.859	1 018	0.856	996
		50	0.948	1 678	0.906	1 807	0.861	1 860	0.860	1 813	0.954	1 146	0.958	1 132
		100	0.992	1 774	0.934	2 052	0.904	2 134	0.895	2 100	**0.988**	1 278	**0.990**	1 245
	opd.	10	0.935	1 575	0.865	1 622	0.792	1 659	0.808	1 621	0.859	1 018	0.856	996
		50	**1.0**	1 610	0.950	1 827	**0.915**	1 916	**0.919**	1 888	0.954	1 146	0.958	1 132
		100	**1.0**	1 610	**0.951**	1 839	**0.915**	1 916	**0.919**	1 888	**0.988**	1 278	**0.990**	1 245

4.4 Candidate Selection

The next step in the transfer pipeline involved selecting documents from the target corpus in order to create new relevance judgments for each query. The goal is to identify documents highly likely to be relevant to a given query. This is crucial, as generating judgments for every query-document pair would be computationally expensive and unnecessary as unjudged documents in a test collection are mostly interpreted as not relevant by default.

To maximize the number of likely relevant documents, three selection approaches were tested. The `Naive` approach retrieved the top 1000 documents from the target corpus using the query text and, when available, a query's narrative for an additional 1000 documents. The `Nearest Neighbor` approach used the top-scoring passages identified in Section 3.3 as independent queries to retrieve the top 20 documents each from the target corpus as candidates. This approach was tested with different numbers of top passages per query: 10, 50, and 100. Additionally, it was tested with and without a restriction that limited each query's top passages to a maximum of one passage per document (`opd.`). The `Union` approach combined all documents retrieved by the `Naive` and each `Nearest Neighbor` approach, filtering out duplicate candidates.

A recall evaluation of the candidate selection approaches is only possible for the relevance transfer within the same dataset, meaning from a source dataset to its own document corpus. This constraint is because existing relevance judgments must be available for the same retrieval task, i.e., for the same queries, in both the source and target datasets. Consequently, Table 4.4 presents recall and the average number of selected candidate documents per query of the transfer pipeline within the same dataset. The recall metric is computed as the proportion of retrieved candidate documents that are judged as relevant relative to the total number of relevant documents in the retrieval task.

The `Naive` approach achieved very high recall for `Touché 20`, benefiting from its relatively small document corpus of 0.4 million documents and the retrieval of up to 2000 documents per query. However, its performance reduces significantly on all other datasets, because of much larger corpora. Additionally, `TREC-19 DL` and `TREC-20 DL` do not provide query narratives, limiting retrieval to a maximum of 1000 documents per query and thereby reducing recall. Overall, the `Naive` approach was able to retrieve a large amount of relevant documents, but this comes at the cost of including many non-relevant candidates, making the selection less efficient.

The `Nearest Neighbor` approach outperformed `Naive` in nearly all variants, except when using only the top 10 passages, limiting candidate documents per query to 200. For `TREC-7` and `TREC-8`, the number of selected documents did not increase between the `opd.` top 50 and top 100 variants due to the document selection process evaluated in Section 4.2. As shown in Table 4.2, these retrieval tasks have only one positive relevance label and are restricted to a maximum of 50 relevant documents per query. Thus, using the top 100 passages did not improve recall, as no additional relevant passages available under the `opd.` constraint. For `Touché 20`, which has two positive relevance labels, a similar effect was observed due to its overall low number of relevance judgments. However, even if `Robust04` averages 36 relevant documents, some queries exceed 50 relevant judgments in total, which explains the higher recall achieved with the `opd. 100` variant. Another expected outcome of Table 4.4 is that the same recall is achieved for `TREC-19 DL` and `TREC-20 DL` for the `def.` and `opd.` variants. This is because their document corpora consist solely of passages, making the restriction of one passage per document redundant. Overall, a key finding is that restricting the top passages to a maximum of one passage per document (`opd.`) consistently improved performance compared to allowing multiple passages per document (`def.`). This result shows enforcing greater document diversity leads to more effective candidate selection.

The Union approach further improved recall, particularly for Robust04, TREC-7, and TREC-8, where it increased recall by approximately 10 percentage points. For TREC-19 DL and TREC-20 DL, the Union opd. 100 variant improved recall by three percentage points compared to Union opd. 50, reaching an impressive recall of approximately 99%. However, this improvement came at the cost of tripling or quadrupling the number of retrieved documents per query. Despite this increase, the trade-off remains acceptable, as recall is consistently high across all six tasks, and the number of retrieved documents per query remains below 2000. Therefore, Union opd. 100 is the most effective candidate selection approach and is used in the subsequent step of the pipeline.

4.5 Pairwise Preferences

The candidate selection process identified Union opd. 100 as the most effective approach for pre-selecting likely relevant documents from the target corpus. Consequently, the evaluation in this section is conducted using the candidate documents selected by this approach. For further processing with large language models, all candidate documents were segmented into passages, following the same procedure as described in Section 3.2.2.

For each query in a retrieval task, all passages from the selected candidate documents were paired with the 15 most relevant and 5 least relevant passages for that query. These 20 passages were determined through the passage-scoring step described in Section 3.3. As evaluated in Section 4.3, the passage scores were derived from the retriever-metric combination with the highest rank correlation, ensuring the selection of the 15 highest-scoring relevant and the 5 lowest-scoring non-relevant passages per query. Since Section 4.4 showed that nearest neighbor approaches performed better when limited to selecting at most one passage per document, this restriction was also applied when selecting the 20 passages, as outlined in Section 3.4.3 under Diversified Selection.

To determine the relevance of candidate documents from the target corpus, tuples consisting of (query, known passage, passage to judge), were processed using various versions of Google's T5 model. For each tuple, the model was prompted to predict the relevance of the target passage to the query, based on the known source passage. After processing all comparisons, each target passage received 20 relevance scores, one for each comparison to a selected source passage. To derive an overall passage score, these individual scores were aggregated and transformed using different methods.

The tested aggregation methods included `mean, min, max, sum`, while the transformation methods included `id, log, exp, sqrt`. Finally, a document's relevance score was determined by taking the maximum of all its passage scores, based on the assumption that a document is only as relevant as its most relevant passage [9].

For comparison, a pointwise approach was tested alongside the pairwise preference approach. In this setup, the `T5` models received only the query and the target passage, without comparisons to known source passages, and was prompted to determine the target passage's relevance to the query. Since this approach evaluated each passage independently, no aggregation or transformation was needed at passage level. However, to derive the final document relevance score, the maximum of all passage scores for the document was used again.

4.5.1 Transfer Pipeline on Source Corpora

The evaluated rank correlation scores between the actual relevance labels of the source datasets and the generated relevance scores from the self-transfer are listed in Table 4.5. These scores represent the average rank correlation across all queries for each retrieval task. The rank correlation for each query was computed based on all candidates from the `Union opd. 100` approach that had a relevance judgment in the corresponding retrieval task.

Overall, the best aggregation method across all three transformer models was `min`, which achieved the highest rank correlation across all retrieval tasks, except one for `t5-small` with `max` aggregation. This result suggests that a passage is only as relevant as its least relevant comparison to the source passages. Additionally, the table shows that applying transformation methods to the aggregated passage scores had no impact on the rank correlation evaluation. This is expected, as tested transformations keep the linear orders of the relevance scores, meaning that applying the same transformation to all aggregated scores has no effect on rank correlation, making the step redundant.

One notable outlier is `Touché 20`, which has two key differences from the other evaluated datasets. First, rank correlation scores using `t5-small` were higher than those for `flan-t5-small` and `flan-t5-base`. Second, the pointwise approach for `flan-t5-base` achieved the overall best rank correlation for `Touché 20`. Unlike the other datasets, `Touché 20` has the smallest document corpus and the fewest relevance judgments per query among all tested retrieval tasks. Additionally, this dataset is known to be noisy and has shown unusual behavior in other analyses as well [27].

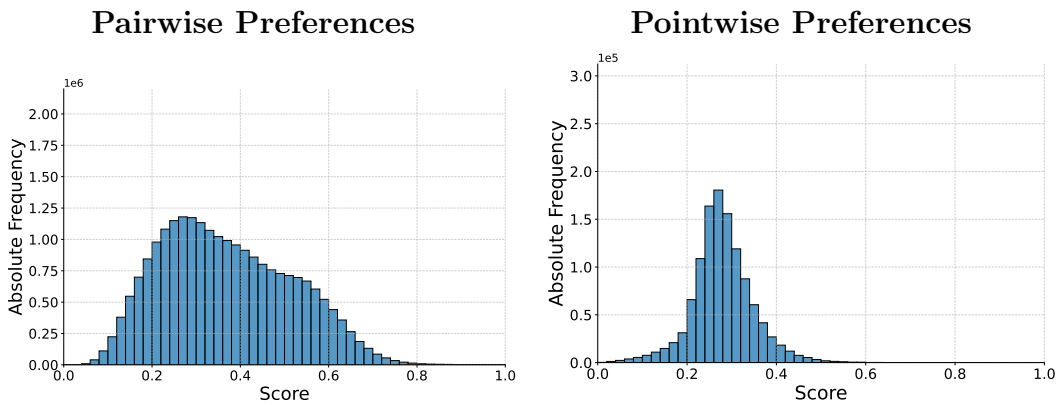
Table 4.5: Rank correlations of the inferred relevance judgements to the original document judgments reported in terms of Kendall's τ and Spearman's ρ.

Approach			Touché 20		Robust04		TREC-7		TREC-8		TREC-19 DL		TREC-20 DL	
			τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ
Pairwise Preferences flan-t5-base	mean	id	0.232	0.288	0.154	0.189	0.126	0.154	0.174	0.213	0.254	0.316	0.233	0.293
		log	0.232	0.288	0.154	0.189	0.126	0.154	0.174	0.213	0.254	0.316	0.233	0.293
		exp	0.232	0.288	0.154	0.189	0.126	0.154	0.174	0.213	0.254	0.316	0.233	0.293
		sqrt	0.232	0.288	0.154	0.189	0.126	0.154	0.174	0.213	0.254	0.316	0.233	0.293
	min	id	**0.250**	**0.314**	**0.169**	**0.208**	**0.137**	**0.167**	**0.187**	**0.229**	**0.311**	**0.387**	**0.293**	**0.366**
		log	**0.250**	**0.314**	**0.169**	**0.208**	**0.137**	**0.167**	**0.187**	**0.229**	**0.311**	**0.387**	**0.293**	**0.366**
		exp	**0.250**	**0.314**	**0.169**	**0.208**	**0.137**	**0.167**	**0.187**	**0.229**	**0.311**	**0.387**	**0.293**	**0.366**
		sqrt	**0.250**	**0.314**	**0.169**	**0.208**	**0.137**	**0.167**	**0.187**	**0.229**	**0.311**	**0.387**	**0.293**	**0.366**
	max	id	0.226	0.283	0.079	0.097	0.084	0.103	0.080	0.098	0.107	0.136	0.120	0.151
		log	0.226	0.283	0.079	0.097	0.084	0.103	0.080	0.098	0.107	0.136	0.120	0.151
		exp	0.226	0.283	0.079	0.097	0.084	0.103	0.080	0.098	0.107	0.136	0.120	0.151
		sqrt	0.226	0.283	0.079	0.097	0.084	0.103	0.080	0.098	0.107	0.136	0.120	0.151
	sum	id	0.232	0.288	0.154	0.189	0.126	0.154	0.174	0.213	0.254	0.316	0.233	0.293
		log	0.232	0.288	0.154	0.189	0.126	0.154	0.174	0.213	0.254	0.316	0.233	0.293
		exp	0.232	0.288	0.154	0.189	0.126	0.154	0.174	0.213	0.254	0.316	0.233	0.293
		sqrt	0.232	0.288	0.154	0.189	0.126	0.154	0.174	0.213	0.254	0.316	0.233	0.293
Pairwise Preferences flan-t5-small	mean	id	0.232	0.299	0.067	0.082	0.065	0.079	0.082	0.101	0.175	0.220	0.132	0.165
		log	0.232	0.299	0.067	0.082	0.065	0.079	0.082	0.101	0.175	0.220	0.132	0.165
		exp	0.232	0.299	0.067	0.082	0.065	0.079	0.082	0.101	0.175	0.220	0.132	0.165
		sqrt	0.232	0.299	0.067	0.082	0.065	0.079	0.082	0.101	0.175	0.220	0.132	0.165
	min	id	**0.301**	**0.388**	**0.082**	**0.101**	**0.080**	**0.097**	**0.094**	**0.115**	**0.224**	**0.282**	**0.179**	**0.224**
		log	**0.301**	**0.388**	**0.082**	**0.101**	**0.080**	**0.097**	**0.094**	**0.115**	**0.224**	**0.282**	**0.179**	**0.224**
		exp	**0.301**	**0.388**	**0.082**	**0.101**	**0.080**	**0.097**	**0.094**	**0.115**	**0.224**	**0.282**	**0.179**	**0.224**
		sqrt	**0.301**	**0.388**	**0.082**	**0.101**	**0.080**	**0.097**	**0.094**	**0.115**	**0.224**	**0.282**	**0.179**	**0.224**
	max	id	0.196	0.250	0.028	0.035	0.030	0.037	0.043	0.052	0.073	0.092	0.048	0.060
		log	0.196	0.250	0.028	0.035	0.030	0.037	0.043	0.052	0.073	0.092	0.048	0.060
		exp	0.196	0.250	0.028	0.035	0.030	0.037	0.043	0.052	0.073	0.092	0.048	0.060
		sqrt	0.196	0.250	0.028	0.035	0.030	0.037	0.043	0.052	0.073	0.092	0.048	0.060
	sum	id	0.232	0.299	0.067	0.082	0.065	0.079	0.082	0.101	0.175	0.220	0.132	0.165
		log	0.232	0.299	0.067	0.082	0.065	0.079	0.082	0.101	0.175	0.220	0.132	0.165
		exp	0.232	0.299	0.067	0.082	0.065	0.079	0.082	0.101	0.175	0.220	0.132	0.165
		sqrt	0.232	0.299	0.067	0.082	0.065	0.079	0.082	0.101	0.175	0.220	0.132	0.165
Pairwise Preferences t5-small	mean	id	0.239	0.308	-0.003	-0.003	0.001	0.001	-0.005	-0.006	-0.050	-0.063	-0.042	-0.053
		log	0.239	0.308	-0.003	-0.003	0.001	0.001	-0.005	-0.006	-0.050	-0.063	-0.042	-0.053
		exp	0.239	0.308	-0.003	-0.003	0.001	0.001	-0.005	-0.006	-0.050	-0.063	-0.042	-0.053
		sqrt	0.239	0.308	-0.003	-0.003	0.001	0.001	-0.005	-0.006	-0.050	-0.063	-0.042	-0.053
	min	id	**0.320**	**0.410**	**0.010**	**0.013**	**0.025**	**0.031**	**0.008**	**0.010**	-0.039	-0.050	**-0.019**	**-0.024**
		log	**0.320**	**0.410**	**0.010**	**0.013**	**0.025**	**0.031**	**0.008**	**0.010**	-0.039	-0.050	**-0.019**	**-0.024**
		exp	**0.320**	**0.410**	**0.010**	**0.013**	**0.025**	**0.031**	**0.008**	**0.010**	-0.039	-0.050	**-0.019**	**-0.024**
		sqrt	**0.320**	**0.410**	**0.010**	**0.013**	**0.025**	**0.031**	**0.008**	**0.010**	-0.039	-0.050	**-0.019**	**-0.024**
	max	id	0.115	0.150	-0.002	-0.002	-0.002	-0.002	-0.004	-0.005	**-0.029**	**-0.037**	-0.042	-0.053
		log	0.115	0.150	-0.002	-0.002	-0.002	-0.002	-0.004	-0.005	**-0.029**	**-0.037**	-0.042	-0.053
		exp	0.115	0.150	-0.002	-0.002	-0.002	-0.002	-0.004	-0.005	**-0.029**	**-0.037**	-0.042	-0.053
		sqrt	0.115	0.150	-0.002	-0.002	-0.002	-0.002	-0.004	-0.005	**-0.029**	**-0.037**	-0.042	-0.053
	sum	id	0.239	0.308	-0.003	-0.003	0.001	0.001	-0.005	-0.006	-0.050	-0.063	-0.042	-0.053
		log	0.239	0.308	-0.003	-0.003	0.001	0.001	-0.005	-0.006	-0.050	-0.063	-0.042	-0.053
		exp	0.239	0.308	-0.003	-0.003	0.001	0.001	-0.005	-0.006	-0.050	-0.063	-0.042	-0.053
		sqrt	0.239	0.308	-0.003	-0.003	0.001	0.001	-0.005	-0.006	-0.050	-0.063	-0.042	-0.053
Pointwise Preferences	flan-t5-base		0.350	0.440	-0.013	-0.016	-0.003	-0.004	-0.026	-0.032	0.037	0.048	0.031	0.039
	flan-t5-small		-0.127	-0.162	-0.011	-0.014	-0.001	-0.001	-0.018	-0.022	-0.048	-0.061	-0.031	-0.040
	t5-small		-0.152	-0.194	0.004	0.005	0.013	0.015	0.005	0.006	-0.056	-0.070	-0.042	-0.053

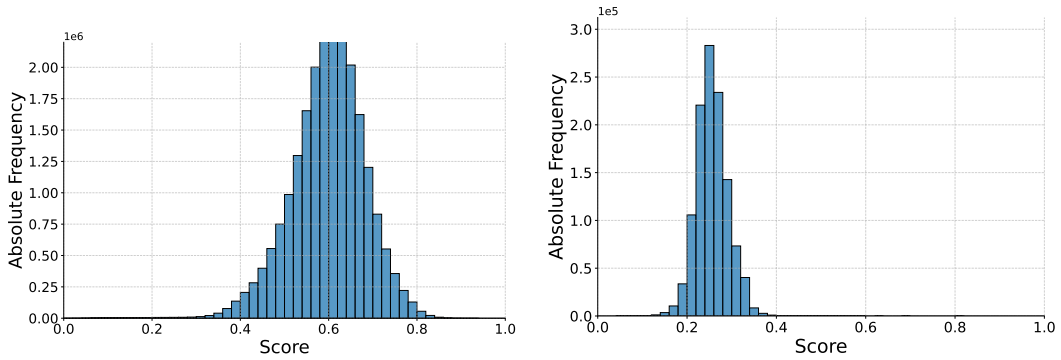
The retrieval tasks based on `Disks4+5` and `MS MARCO` produced highly acceptable results. In all cases, the most advanced transformer model, `flan-t5-base`, outperformed all other models using the pairwise preference approach. A key finding from this evaluation is that the pointwise preference approach performed poorly across all tested models, showing almost no rank correlation between predicted relevance and actual labels. This highlights the limitation of pointwise approaches in predicting document relevance based solely on the document itself. In contrast, the pairwise preference approach achieved notable rank correlation values. Specifically, for `flan-t5-base`, the retrieval tasks based on the `Disks4+5` corpus reached rank correlations of up to 0.229. For the retrieval tasks based on the `MS MARCO` passage corpus, the inferred relevance scores achieved rank correlations of approximately 0.3 for Kendall's τ and 3.7 for Spearman's ρ, indicating a weak to moderate rank correlation.

The high correlation scores for `TREC-DL 19` and `TREC-DL 20`, when compared to the other datasets, except `Touché 20`, are likely due to two factors. First, as analyzed in Section 4.2, these retrieval tasks have a relatively high number of judgments per query, providing a strong foundation for relevance transfer. Second, the `MS MARCO` corpus is a passage corpus, meaning all documents processed by the transfer pipeline were already given as judged passages. As a result, the pairwise comparisons performed the evaluation on individual passages for `MS MARCO` rather than segmented sections of longer documents. Additionally, the segmentation using `spaCy` segments text based on punctuation and maximum passage length which is more rudimentary. Since this work does not include an evaluation of document segmentation methods, the question of optimal document segmentation remains open for future research.

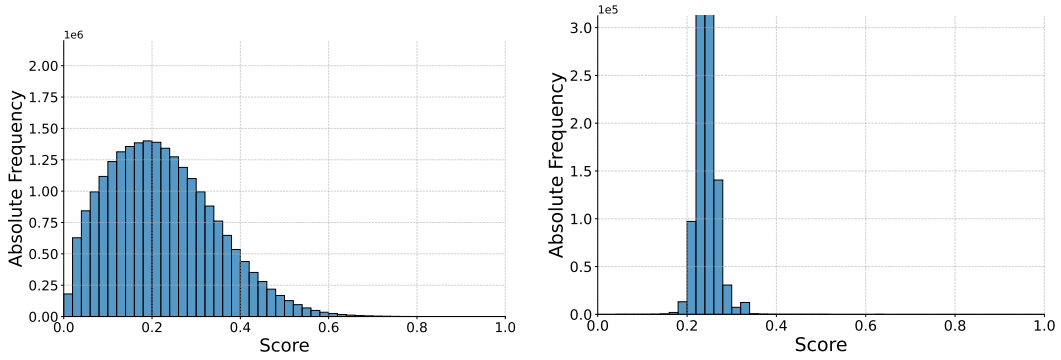
The evaluation was also conducted using the `greedy` versions of Kendall's and Spearman's rank correlations, as shown in Table A.2. Across all datasets, these scores were slightly higher, but the trends observed with the default Kendall and Spearman metrics remained consistent.



(a) Relevance scores generated using `flan-t5-base`.



(b) Relevance scores generated using `flan-t5-small`.



(c) Relevance scores generated using `t5-small`.

Figure 4.1: Distributions of the inferred passage relevance scores for self-transfer across all six retrieval tasks, using the pairwise and pointwise approaches with different versions of the `T5` model.

To better analyze the rank correlation results, Figure 4.1 visualizes the distribution of inferred relevance scores for both the pairwise and pointwise approaches across all six retrieval tasks. In general, the pairwise approaches have greater score variance than the pointwise approaches. This is likely because the pointwise models generate only one relevance score per passage, whereas the pairwise models compare each passage against 20 source passages, leading to more granular relevance differentiation. The comparison to 20 source passages rather than solely on the target passage itself also explains the overall higher number of scores in the pairwise approaches.

When comparing pointwise models, it is evident that more advanced models introduce greater variance into the score distribution. However, the pairwise approaches exhibit significantly higher score variance overall. Looking at the distribution of the pairwise models, notable differences emerge. While `t5-small` tends to underestimate relevance scores, `flan-t5-small` assigns generally higher scores on average. In contrast, `flan-t5-base` produces the highest score variance, allowing for more effective passage differentiation. Additionally, unlike `flan-t5-small` or `t5-small`, its score distribution is more spread out rather than forming a sharp peak, suggesting a more nuanced relevance assessment. This broader score distribution is a key indicator why `flan-t5-base` achieved the best overall results for both `default` and `greedy` rank correlation. In conclusion, the ability to infer a broad spectrum of relevance scores seems to allow for better passage differentiation, reinforcing the idea that a passage's overall relevance is dependent on its weakest passage regarding that the `min` aggregation performed best.

Table 4.6: Empirically selected queries from the six retrieval tasks used in the transfer pipeline to generate new relevance judgments for `ClueWeb22/b`.

Dataset	Query ID	Query Text
Touché 20	34	„Are social networking sites good for our society?“
	49	„Should body cameras be mandatory for police?“
Robust04	448	„ship losses“
	681	„wind power location“
TREC-7	354	„journalist risks“
	358	„blood-alcohol fatalities“
TREC-8	422	„art, stolen, forged“
	441	„Lyme disease“
TREC-19 DL	1037798	„who is robert gray“
	1129237	„hydrogen is a liquid below what temperature“
TREC-20 DL	997622	„where is the show shameless filmed“
	1051399	„who sings monk theme song“
	1127540	„meaning of shebang“

4.5.2 Transfer Pipeline on `ClueWeb22/b`

The transfer pipeline to `ClueWeb22/b` followed the same setup as the self-transfer evaluation. However, a key challenge in this transfer was the absence of existing relevance judgments for the queries of the six tested retrieval tasks needed for evaluation. To address this issue, two to three queries from each retrieval task were manually selected, as shown in Table 4.6. The relevance transfer pipeline was then applied to generate new relevance judgments for documents from the `ClueWeb22/b` corpus for these queries.

Since the `Union opd. 100` candidate selection was identified as the most effective approach in Section 4.4, it was again used to determine candidate documents within the `ClueWeb22/b` corpus. After selecting the candidate documents for the chosen queries, a pooling process was conducted. Twelve lexical models[1] and three neural models[2] were utilized. The 15 pooled retrieval models were executed in their archived version from `TIRA/TIREx` [12, 13]. From that pooling, a top-10 pool of the 15 systems was created using trectools [23]. The remaining documents after pooling were then manually assessed to determine their relevance to the corresponding queries.

[1] BM25, DFIC, DFIZ, DirichletLM, DFRee, DLH, DPH, Hiemstra_LM, InB2, LGD, Js_KLs, PL2

[2] ANCE Base Cosine, MonoT5 Base, MonoBERT Base

For this assessment, the top three passages of each pooled document, determined by the inferred relevance scores from the pairwise preference approach using `flan-t5-base`, were presented to a human judge. The judge then assigned a relevance label of 0, 1, or 2 to each passage, where 0 indicated non-relevance, 1 indicated relevance, and 2 indicated high relevance. The highest relevance label among the three passages was then assigned as the document's relevance judgment. These relevance judgments were used to evaluate the rank correlation between the inferred relevance scores and the manually assessed relevance labels for `ClueWeb22/b`.

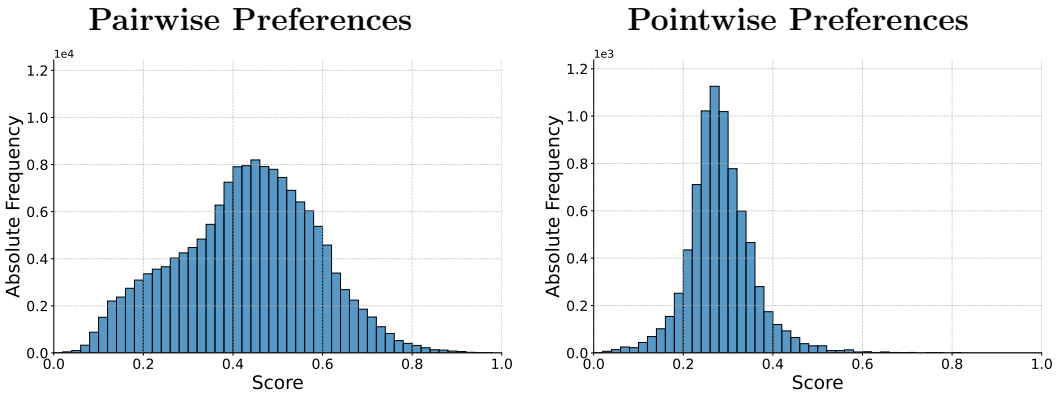
The rank correlation scores between the inferred relevance judgments and the manually assessed relevance labels are listed in Table 4.7. The transfer process to `ClueWeb22/b` exhibited similar behavior to the self-transfer results in Table 4.5. Additionally, the distribution of inferred relevance scores for the pooled candidates, shown in Figure 4.2, closely matched the self-transfer distribution in Figure 4.1. Again, the pairwise preference approach using `flan-t5-base` achieved the highest rank correlation scores, yielding moderate to good correlations. Despite some outliers, the `min` aggregation method once again proved to be the most effective. Unlike previous observations, `Touché 20` aligned with the other retrieval tasks in this evaluation, showing no unusual behavior. However, `Robust04` stood out as an outlier, as the pointwise approach with `t5-small` unexpectedly achieved the highest rank correlation, while all other approaches overall exhibited little to no rank correlation.

During the annotation process for `Robust04`, it was observed that relevant documents did not transfer well from its news domain to the general-purpose web domain of `ClueWeb22/b`, as only a few documents were judged as relevant. This is likely due to the brevity of query 448 and 681, which consist of only two or three words, creating a significant gap between the query text and its narrative. Since the pairwise preference approach relied solely on the query text, while the annotator had access to the full description and narrative, this lack of contextual information likely impacted the model's ability to infer relevance accurately. A potential improvement could be to incorporate the description or narrative into the `DuoPrompt` to provide the model with a richer understanding of the query, allowing for more precise relevance judgments.

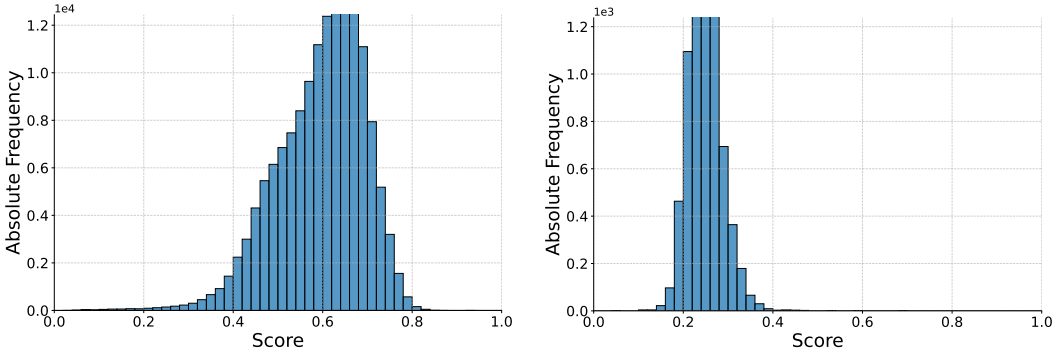
Table A.3 presents the rank correlation scores for the `greedy` versions of Kendall's τ and Spearman's ρ. The greedy mapping of the inferred relevance scores to the relevance labels 0, 1, or 2 achieved strong agreement in both pairwise and pointwise, with the manually created judgments. The pairwise approach still outperforms the pointwise approach.

Table 4.7: Rank correlations between inferred relevance judgments and manually created judgments for `ClueWeb22/b`, reported using Kendall's τ and Spearman's ρ.

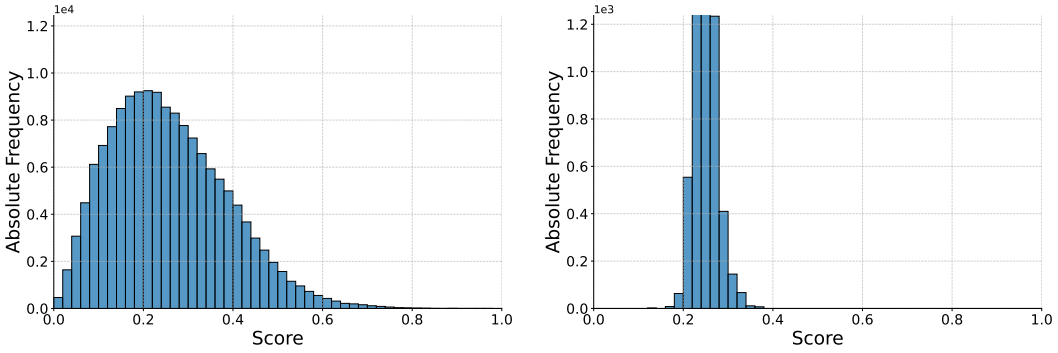
Approach			Touché 20		Robust04		TREC-7		TREC-8		TREC-19 DL		TREC-20 DL	
			τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ
Pairwise Preferences flan-t5-base	mean	id	**0.330**	0.411	-0.098	-0.114	0.273	0.342	0.132	0.166	0.305	0.389	0.052	0.059
		log	**0.330**	0.411	-0.098	-0.114	0.273	0.342	0.132	0.166	0.305	0.389	0.052	0.059
		exp	**0.330**	0.411	-0.098	-0.114	0.273	0.342	0.132	0.166	0.305	0.389	0.052	0.059
		sqrt	**0.330**	0.411	-0.098	-0.114	0.273	0.342	0.132	0.166	0.305	0.389	0.052	0.059
	min	id	0.329	**0.417**	**-0.080**	**-0.096**	**0.298**	**0.363**	**0.263**	**0.322**	**0.336**	**0.424**	0.156	0.183
		log	0.329	**0.417**	**-0.080**	**-0.096**	**0.298**	**0.363**	**0.263**	**0.322**	**0.336**	**0.424**	0.156	0.183
		exp	0.329	**0.417**	**-0.080**	**-0.096**	**0.298**	**0.363**	**0.263**	**0.322**	**0.336**	**0.424**	0.156	0.183
		sqrt	0.329	**0.417**	**-0.080**	**-0.096**	**0.298**	**0.363**	**0.263**	**0.322**	**0.336**	**0.424**	0.156	0.183
	max	id	0.038	0.046	-0.165	-0.210	0.239	0.291	0.105	0.135	0.072	0.090	**0.159**	**0.201**
		log	0.038	0.046	-0.165	-0.210	0.239	0.291	0.105	0.135	0.072	0.090	**0.159**	**0.201**
		exp	0.038	0.046	-0.165	-0.210	0.239	0.291	0.105	0.135	0.072	0.090	**0.159**	**0.201**
		sqrt	0.038	0.046	-0.165	-0.210	0.239	0.291	0.105	0.135	0.072	0.090	**0.159**	**0.201**
	sum	id	**0.330**	0.411	-0.098	-0.114	0.273	0.342	0.132	0.166	0.305	0.389	0.052	0.059
		log	**0.330**	0.411	-0.098	-0.114	0.273	0.342	0.132	0.166	0.305	0.389	0.052	0.059
		exp	**0.330**	0.411	-0.098	-0.114	0.273	0.342	0.132	0.166	0.305	0.389	0.052	0.059
		sqrt	**0.330**	0.411	-0.098	-0.114	0.273	0.342	0.132	0.166	0.305	0.389	0.052	0.059
Pairwise Preferences flan-t5-small	mean	id	0.063	0.073	-0.192	-0.232	0.306	0.375	0.071	0.090	0.316	0.387	**0.212**	**0.269**
		log	0.063	0.073	-0.192	-0.232	0.306	0.375	0.071	0.090	0.316	0.387	**0.212**	**0.269**
		exp	0.063	0.073	-0.192	-0.232	0.306	0.375	0.071	0.090	0.316	0.387	**0.212**	**0.269**
		sqrt	0.063	0.073	-0.192	-0.232	0.306	0.375	0.071	0.090	0.316	0.387	**0.212**	**0.269**
	min	id	**0.142**	**0.176**	**-0.167**	**-0.205**	**0.376**	**0.468**	**0.169**	**0.219**	**0.338**	**0.407**	0.161	0.210
		log	**0.142**	**0.176**	**-0.167**	**-0.205**	**0.376**	**0.468**	**0.169**	**0.219**	**0.338**	**0.407**	0.161	0.210
		exp	**0.142**	**0.176**	**-0.167**	**-0.205**	**0.376**	**0.468**	**0.169**	**0.219**	**0.338**	**0.407**	0.161	0.210
		sqrt	**0.142**	**0.176**	**-0.167**	**-0.205**	**0.376**	**0.468**	**0.169**	**0.219**	**0.338**	**0.407**	0.161	0.210
	max	id	-0.049	-0.063	-0.187	-0.220	0.146	0.180	0.009	0.012	0.177	0.217	0.125	0.162
		log	-0.049	-0.063	-0.187	-0.220	0.146	0.180	0.009	0.012	0.177	0.217	0.125	0.162
		exp	-0.049	-0.063	-0.187	-0.220	0.146	0.180	0.009	0.012	0.177	0.217	0.125	0.162
		sqrt	-0.049	-0.063	-0.187	-0.220	0.146	0.180	0.009	0.012	0.177	0.217	0.125	0.162
	sum	id	0.063	0.073	-0.192	-0.232	0.306	0.375	0.071	0.090	0.316	0.387	**0.212**	**0.269**
		log	0.063	0.073	-0.192	-0.232	0.306	0.375	0.071	0.090	0.316	0.387	**0.212**	**0.269**
		exp	0.063	0.073	-0.192	-0.232	0.306	0.375	0.071	0.090	0.316	0.387	**0.212**	**0.269**
		sqrt	0.063	0.073	-0.192	-0.232	0.306	0.375	0.071	0.090	0.316	0.387	**0.212**	**0.269**
Pairwise Preferences t5-small	mean	id	**0.074**	**0.105**	-0.001	0.001	0.094	0.105	-0.052	-0.067	-0.026	-0.037	0.100	0.130
		log	**0.074**	**0.105**	-0.001	0.001	0.094	0.105	-0.052	-0.067	-0.026	-0.037	0.100	0.130
		exp	**0.074**	**0.105**	-0.001	0.001	0.094	0.105	-0.052	-0.067	-0.026	-0.037	0.100	0.130
		sqrt	**0.074**	**0.105**	-0.001	0.001	0.094	0.105	-0.052	-0.067	-0.026	-0.037	0.100	0.130
	min	id	0.059	0.073	**0.132**	**0.162**	**0.117**	**0.134**	-0.032	-0.038	-0.029	-0.029	0.052	0.067
		log	0.059	0.073	**0.132**	**0.162**	**0.117**	**0.134**	-0.032	-0.038	-0.029	-0.029	0.052	0.067
		exp	0.059	0.073	**0.132**	**0.162**	**0.117**	**0.134**	-0.032	-0.038	-0.029	-0.029	0.052	0.067
		sqrt	0.059	0.073	**0.132**	**0.162**	**0.117**	**0.134**	-0.032	-0.038	-0.029	-0.029	0.052	0.067
	max	id	0.063	0.099	0.025	0.035	0.107	0.129	**0.049**	**0.054**	**0.069**	**0.083**	**0.109**	**0.139**
		log	0.063	0.099	0.025	0.035	0.107	0.129	**0.049**	**0.054**	**0.069**	**0.083**	**0.109**	**0.139**
		exp	0.063	0.099	0.025	0.035	0.107	0.129	**0.049**	**0.054**	**0.069**	**0.083**	**0.109**	**0.139**
		sqrt	0.063	0.099	0.025	0.035	0.107	0.129	**0.049**	**0.054**	**0.069**	**0.083**	**0.109**	**0.139**
	sum	id	**0.074**	**0.105**	-0.001	0.001	0.094	0.105	-0.052	-0.067	-0.026	-0.037	0.100	0.130
		log	**0.074**	**0.105**	-0.001	0.001	0.094	0.105	-0.052	-0.067	-0.026	-0.037	0.100	0.130
		exp	**0.074**	**0.105**	-0.001	0.001	0.094	0.105	-0.052	-0.067	-0.026	-0.037	0.100	0.130
		sqrt	**0.074**	**0.105**	-0.001	0.001	0.094	0.105	-0.052	-0.067	-0.026	-0.037	0.100	0.130
Pointwise Preferences	flan-t5-base		0.131	0.174	-0.046	-0.058	-0.017	-0.020	0.033	0.038	0.077	0.091	0.207	0.255
	flan-t5-small		0.101	0.131	-0.043	-0.056	0.021	0.021	0.110	0.139	0.088	0.103	0.101	0.134
	t5-small		0.183	0.227	0.134	0.166	0.063	0.074	0.095	0.119	0.119	0.152	0.122	0.165



(a) Relevance scores generated using `flan-t5-base`.



(b) Relevance scores generated using `flan-t5-small`.



(c) Relevance scores generated using `t5-small`.

Figure 4.2: Distributions of the inferred passage relevance scores for the pooled documents transferred to `ClueWeb22/b` across all six retrieval tasks, using the pairwise and pointwise approaches with different versions of the `T5` model.

Chapter 5

Conclusion & Future Work

This thesis presented an approach to automatically generate relevance judgments using an existing annotated test collection. This collection consisted of a document corpus and a retrieval task with predefined queries and relevance judgments. To enable the transfer of relevance judgments to another document corpus, a transfer pipeline based on pairwise comparisons was developed. The evaluation of the pipeline showed that the pairwise preference approach significantly outperforms a pointwise approach, both in a self-transfer setting and in the generation of relevance judgments for `ClueWeb22/b`. This indicates that comparing an unjudged document with already judged documents improves the quality of automatically generated judgments. However, successful transfer requires a source dataset with a sufficient number of high-quality relevance judgments. While the pairwise preference approach shows great potential for reducing the manual effort required to generate relevance judgments, it does not yet achieve the same quality as human judgments. Therefore, manual judgments are still necessary, and automatic generation is currently best suited for pre-judgment tasks rather than final relevance judgments.

The comparison of different `T5` model variants showed that more fine-tuned models achieved higher rank correlations. It is therefore likely that a larger version of `FLAN-T5` with more parameters would yield even better relevance predictions. Future work could explore alternative large language models and improved prompting strategies, such as incorporating a query's description or narrative, for pairwise preference inference. Another key component of the transfer pipeline was the `Union opd. 100` approach, which proved to be the most effective candidate selection method. However, refining candidate selection remains an open research task. The nearest-neighbor approach already showed strong performance, and its combination with the naive selection method further increased recall, albeit at the cost of a larger candidate set.

Appendix A

Greedy Evaluation

Table A.1: Average rank correlations between the assigned passage scores and the original relevance judgments across all queries of a retrieval task, reported using the `greedy` version of Kendall's τ and Spearman's ρ.

	Retrieval Model	Touché 20		Robust04		TREC-7		TREC-8		TREC-19 DL		TREC-20 DL		Avg.	
		τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ
nDCG@10	BM25	0.886	0.908	0.951	0.957	0.954	0.954	0.971	0.971	0.800	0.837	0.843	0.874	0.901	0.917
	DFR_BM25	0.886	0.909	0.950	0.956	0.952	0.952	0.972	0.972	0.803	0.840	0.845	0.877	0.901	0.918
	DFIZ	0.865	0.888	0.953	0.959	0.957	0.957	0.970	0.970	0.805	0.841	0.839	0.871	0.898	0.914
	DLH	0.912	0.932	0.950	0.956	0.954	0.954	0.973	0.973	0.804	0.841	0.840	0.872	0.905	0.921
	DPH	0.866	0.890	0.951	0.957	0.955	0.955	0.971	0.971	0.794	0.832	0.838	0.869	0.896	0.912
	DirichletLM	0.756	0.781	0.951	0.958	0.958	0.958	0.966	0.966	0.790	0.826	0.824	0.857	0.874	0.891
	Hiemstra_LM	0.923	0.944	0.952	0.958	0.956	0.956	0.971	0.971	0.805	0.841	0.842	0.874	**0.908**	**0.924**
	LGD	0.890	0.914	0.952	0.958	0.959	0.959	0.968	0.968	0.806	0.842	0.843	0.874	0.903	0.919
	PL2	0.888	0.912	0.948	0.954	0.952	0.952	0.973	0.973	0.798	0.835	0.849	0.880	0.901	0.918
	TF_IDF	0.892	0.916	0.949	0.955	0.953	0.953	0.969	0.969	0.803	0.840	0.844	0.876	0.902	0.918
precision@10	BM25	0.797	0.830	0.880	0.890	0.878	0.878	0.901	0.901	0.693	0.744	0.732	0.776	0.814	0.837
	DFR_BM25	0.801	0.833	0.879	0.889	0.877	0.877	0.902	0.902	0.692	0.743	0.733	0.777	0.814	0.837
	DFIZ	0.768	0.801	0.876	0.886	0.876	0.876	0.889	0.889	0.699	0.748	0.732	0.777	0.807	0.829
	DLH	0.829	0.864	0.890	0.900	0.887	0.887	0.924	0.924	0.700	0.750	0.733	0.777	0.827	0.850
	DPH	0.788	0.822	0.876	0.886	0.872	0.872	0.893	0.893	0.690	0.740	0.731	0.775	0.808	0.831
	DirichletLM	0.687	0.716	0.867	0.877	0.869	0.869	0.879	0.879	0.683	0.731	0.735	0.778	0.787	0.808
	Hiemstra_LM	0.845	0.883	0.887	0.897	0.885	0.885	0.916	0.916	0.698	0.746	0.738	0.782	**0.828**	**0.851**
	LGD	0.804	0.840	0.873	0.883	0.873	0.873	0.882	0.882	0.695	0.744	0.735	0.778	0.810	0.834
	PL2	0.805	0.838	0.883	0.893	0.886	0.886	0.917	0.917	0.688	0.738	0.734	0.777	0.819	0.842
	TF_IDF	0.801	0.834	0.881	0.891	0.878	0.878	0.902	0.902	0.694	0.745	0.732	0.775	0.815	0.837

Table A.2: Rank correlations of the inferred relevance judgments to the original relevance judgments for self-transfer, reported using the `greedy` version of Kendall's τ and Spearman's ρ.

Approach			Touché 20		Robust04		TREC-7		TREC-8		TREC-19 DL		TREC-20 DL	
			τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ
Pairwise Preferences flan-t5-base	mean	id	0.387	0.413	0.304	0.304	0.279	0.279	0.317	0.317	0.363	0.382	0.411	0.429
		log	0.384	0.410	0.304	0.305	0.277	0.277	0.317	0.317	0.350	0.369	0.410	0.428
		exp	0.384	0.410	0.304	0.305	0.280	0.280	0.318	0.318	0.359	0.378	0.412	0.430
		sqrt	0.386	0.412	0.304	0.305	0.278	0.278	0.317	0.317	0.354	0.374	0.417	0.436
	min	id	**0.393**	**0.419**	**0.342**	**0.343**	**0.295**	**0.295**	**0.351**	**0.351**	**0.402**	**0.425**	**0.456**	**0.476**
		log	**0.385**	**0.412**	**0.340**	**0.341**	**0.294**	**0.294**	**0.349**	**0.349**	**0.404**	**0.428**	**0.453**	**0.474**
		exp	**0.391**	**0.417**	**0.343**	**0.344**	**0.296**	**0.296**	**0.352**	**0.352**	**0.405**	**0.428**	**0.453**	**0.473**
		sqrt	**0.386**	**0.412**	**0.341**	**0.342**	**0.295**	**0.295**	**0.350**	**0.350**	**0.405**	**0.428**	**0.455**	**0.475**
	max	id	0.360	0.387	0.144	0.144	0.159	0.159	0.153	0.153	0.211	0.223	0.272	0.282
		log	0.357	0.383	0.144	0.145	0.158	0.158	0.151	0.151	0.213	0.225	0.269	0.279
		exp	0.356	0.383	0.144	0.144	0.158	0.158	0.152	0.152	0.209	0.220	0.270	0.280
		sqrt	0.355	0.382	0.144	0.145	0.158	0.158	0.152	0.152	0.213	0.225	0.269	0.278
	sum	id	0.387	0.413	0.304	0.304	0.279	0.279	0.317	0.317	0.363	0.382	0.411	0.429
		log	0.384	0.410	0.304	0.305	0.277	0.277	0.317	0.317	0.350	0.369	0.410	0.428
		exp	0.302	0.323	0.289	0.290	0.264	0.264	0.304	0.304	0.328	0.346	0.390	0.406
		sqrt	0.386	0.412	0.304	0.305	0.278	0.278	0.317	0.317	0.354	0.374	0.417	0.436
Pairwise Preferences flan-t5-small	mean	id	0.366	0.395	0.170	0.171	0.170	0.170	0.193	0.193	0.269	0.283	0.249	0.258
		log	0.366	0.393	0.170	0.171	0.169	0.169	0.194	0.194	0.266	0.280	0.249	0.258
		exp	0.364	0.394	0.171	0.171	0.170	0.170	0.195	0.195	0.264	0.278	0.251	0.261
		sqrt	0.365	0.392	0.171	0.171	0.169	0.169	0.193	0.193	0.269	0.283	0.251	0.261
	min	id	**0.448**	**0.482**	**0.206**	**0.207**	**0.197**	**0.197**	**0.225**	**0.225**	**0.333**	**0.351**	**0.298**	**0.311**
		log	**0.444**	**0.478**	**0.203**	**0.203**	**0.195**	**0.195**	**0.220**	**0.220**	**0.331**	**0.348**	**0.306**	**0.319**
		exp	**0.445**	**0.480**	**0.207**	**0.208**	**0.197**	**0.197**	**0.225**	**0.225**	**0.334**	**0.352**	**0.302**	**0.315**
		sqrt	**0.444**	**0.478**	**0.205**	**0.206**	**0.196**	**0.196**	**0.225**	**0.225**	**0.335**	**0.353**	**0.304**	**0.317**
	max	id	0.324	0.349	0.109	0.110	0.115	0.115	0.128	0.128	0.131	0.138	0.123	0.127
		log	0.328	0.353	0.110	0.110	0.114	0.114	0.127	0.127	0.130	0.138	0.122	0.126
		exp	0.326	0.350	0.109	0.110	0.115	0.115	0.129	0.129	0.132	0.140	0.122	0.126
		sqrt	0.323	0.348	0.109	0.110	0.114	0.114	0.129	0.129	0.132	0.139	0.122	0.126
	sum	id	0.366	0.395	0.170	0.171	0.170	0.170	0.193	0.193	0.269	0.283	0.249	0.258
		log	0.366	0.393	0.170	0.171	0.169	0.169	0.194	0.194	0.266	0.280	0.249	0.258
		exp	0.369	0.397	0.171	0.172	0.167	0.167	0.194	0.194	0.266	0.280	0.252	0.262
		sqrt	0.365	0.392	0.171	0.171	0.169	0.169	0.193	0.193	0.269	0.283	0.251	0.261
Pairwise Preferences t5-small	mean	id	0.379	0.409	0.070	0.070	0.084	0.084	0.085	0.085	-0.002	-0.002	0.011	0.010
		log	0.379	0.409	0.069	0.069	0.084	0.084	0.083	0.083	-0.005	-0.006	0.012	0.010
		exp	0.378	0.409	0.069	0.070	0.085	0.085	0.084	0.084	0.002	0.001	0.012	0.010
		sqrt	0.380	0.410	0.070	0.070	0.084	0.084	0.084	0.084	-0.004	-0.004	0.009	0.007
	min	id	**0.470**	**0.510**	0.081	0.081	**0.100**	**0.100**	**0.092**	**0.092**	0.028	0.030	0.045	0.046
		log	**0.466**	**0.507**	0.080	0.080	**0.101**	**0.101**	**0.093**	**0.093**	0.026	0.028	0.044	0.045
		exp	**0.465**	**0.505**	0.080	0.080	**0.099**	**0.099**	**0.092**	**0.092**	0.027	0.029	0.043	0.044
		sqrt	**0.472**	**0.513**	0.081	0.081	**0.101**	**0.101**	**0.093**	**0.093**	0.027	0.029	0.044	0.044
	max	id	0.284	0.307	**0.086**	**0.086**	0.090	0.090	0.087	0.087	**0.035**	**0.037**	**0.055**	**0.057**
		log	0.280	0.302	**0.083**	**0.083**	0.089	0.089	0.084	0.084	**0.032**	**0.033**	**0.056**	**0.057**
		exp	0.278	0.300	**0.085**	**0.086**	0.091	0.091	0.087	0.087	**0.036**	**0.037**	**0.060**	**0.062**
		sqrt	0.283	0.306	**0.085**	**0.085**	0.091	0.091	0.086	0.086	**0.032**	**0.034**	**0.057**	**0.058**
	sum	id	0.379	0.409	0.070	0.070	0.084	0.084	0.085	0.085	-0.002	-0.002	0.011	0.010
		log	0.379	0.409	0.069	0.069	0.084	0.084	0.083	0.083	-0.005	-0.006	0.012	0.010
		exp	0.334	0.361	0.058	0.059	0.061	0.061	0.065	0.065	0.031	0.033	0.059	0.061
		sqrt	0.380	0.410	0.070	0.070	0.084	0.084	0.084	0.084	-0.004	-0.004	0.009	0.007
Pointwise Preferences	flan-t5-base		0.497	0.538	0.062	0.062	0.082	0.082	0.075	0.075	0.108	0.114	0.098	0.103
	flan-t5-small		0.065	0.064	0.060	0.060	0.082	0.082	0.075	0.075	0.028	0.029	0.062	0.064
	t5-small		-0.025	-0.035	0.069	0.070	0.082	0.082	0.083	0.083	0.011	0.011	0.006	0.006

Table A.3: Rank correlations of the inferred relevance judgements to the manually created judgments for `ClueWeb22/b` reported in terms of the `greedy` version of Kendall's τ and Spearman's ρ.

Approach			Touché 20		Robust04		TREC-7		TREC-8		TREC-19 DL		TREC-20 DL	
			τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ
Pairwise Preferences flan-t5-base	mean	id	0.462	0.476	0.121	0.121	0.401	0.417	0.335	0.345	0.413	0.434	0.234	0.238
		log	0.462	0.476	0.094	0.096	0.385	0.402	0.335	0.345	0.413	0.434	0.234	0.238
		exp	0.462	0.476	0.075	0.075	0.418	0.435	0.327	0.337	0.413	0.434	0.234	0.238
		sqrt	0.462	0.476	0.075	0.075	0.401	0.417	0.327	0.337	0.413	0.434	0.234	0.238
	min	id	**0.503**	**0.520**	**0.275**	**0.279**	**0.418**	**0.435**	**0.363**	**0.377**	**0.471**	**0.485**	**0.353**	**0.361**
		log	**0.503**	**0.520**	**0.238**	**0.242**	**0.418**	**0.435**	**0.363**	**0.377**	**0.471**	**0.485**	**0.394**	**0.393**
		exp	**0.503**	**0.520**	**0.283**	**0.287**	**0.418**	**0.435**	**0.363**	**0.377**	**0.471**	**0.485**	**0.353**	**0.361**
		sqrt	**0.503**	**0.520**	**0.283**	**0.287**	**0.418**	**0.435**	**0.363**	**0.377**	**0.471**	**0.485**	**0.353**	**0.361**
	max	id	0.356	0.363	-0.123	-0.128	0.313	0.318	0.237	0.250	0.158	0.162	0.332	0.343
		log	0.330	0.340	-0.204	-0.211	0.313	0.318	0.237	0.250	0.158	0.162	0.345	0.356
		exp	0.356	0.363	-0.123	-0.128	0.313	0.318	0.237	0.250	0.158	0.162	0.307	0.317
		sqrt	0.356	0.363	-0.123	-0.128	0.311	0.318	0.237	0.250	0.158	0.162	0.345	0.356
	sum	id	0.462	0.476	0.121	0.121	0.401	0.417	0.335	0.345	0.413	0.434	0.234	0.238
		log	0.462	0.476	0.094	0.096	0.385	0.402	0.335	0.345	0.413	0.434	0.234	0.238
		exp	0.462	0.476	0.257	0.260	0.398	0.417	0.327	0.337	0.401	0.421	0.261	0.270
		sqrt	0.462	0.476	0.075	0.075	0.401	0.417	0.327	0.337	0.413	0.434	0.234	0.238
Pairwise Preferences flan-t5-small	mean	id	0.187	0.189	-0.292	-0.300	0.413	0.428	0.243	0.256	0.517	0.530	**0.425**	**0.440**
		log	0.187	0.189	-0.292	-0.300	0.413	0.428	0.243	0.256	0.517	0.530	0.380	0.394
		exp	0.187	0.189	-0.292	-0.300	0.413	0.428	0.254	0.269	0.517	0.530	**0.403**	**0.418**
		sqrt	0.187	0.189	-0.292	-0.300	0.413	0.428	0.243	0.256	0.517	0.530	**0.444**	**0.460**
	min	id	**0.300**	**0.303**	**-0.280**	**-0.287**	**0.451**	**0.476**	**0.280**	**0.298**	**0.545**	**0.552**	0.358	0.376
		log	**0.300**	**0.303**	**-0.267**	**-0.274**	**0.451**	**0.476**	**0.280**	**0.298**	**0.547**	**0.557**	**0.391**	**0.411**
		exp	**0.300**	**0.303**	**-0.280**	**-0.287**	**0.468**	**0.494**	**0.280**	**0.298**	**0.545**	**0.552**	0.391	0.411
		sqrt	**0.300**	**0.303**	**-0.280**	**-0.287**	**0.451**	**0.476**	**0.280**	**0.296**	**0.545**	**0.552**	0.391	0.411
	max	id	0.098	0.101	-0.315	-0.325	0.271	0.279	0.116	0.122	0.300	0.309	0.257	0.270
		log	0.126	0.129	-0.303	-0.313	0.259	0.267	0.116	0.122	0.303	0.313	0.264	0.277
		exp	0.098	0.101	-0.315	-0.325	0.244	0.252	0.116	0.122	0.306	0.314	0.264	0.277
		sqrt	0.098	0.101	-0.315	-0.325	0.259	0.267	0.116	0.122	0.295	0.304	0.257	0.270
	sum	id	0.187	0.189	-0.292	-0.300	0.413	0.428	0.243	0.256	0.517	0.530	**0.425**	**0.440**
		log	0.187	0.189	-0.292	-0.300	0.413	0.428	0.243	0.256	0.517	0.530	0.380	0.394
		exp	0.205	0.207	-0.292	-0.300	0.413	0.428	0.243	0.258	0.528	0.535	0.380	0.394
		sqrt	0.187	0.189	-0.292	-0.300	0.413	0.428	0.243	0.256	0.517	0.530	**0.444**	**0.460**
Pairwise Preferences t5-small	mean	id	**0.460**	**0.482**	0.029	0.029	0.224	0.231	-0.058	-0.062	0.132	0.135	**0.285**	**0.302**
		log	**0.460**	**0.482**	0.029	0.029	0.224	0.231	-0.058	-0.062	0.142	0.146	**0.285**	**0.302**
		exp	**0.460**	**0.482**	0.029	0.029	0.267	0.273	-0.068	-0.072	0.142	0.146	**0.285**	**0.302**
		sqrt	**0.460**	**0.482**	0.029	0.029	0.212	0.218	-0.048	-0.052	0.132	0.135	**0.285**	**0.302**
	min	id	0.262	0.278	**0.366**	**0.379**	0.222	0.228	-0.008	-0.010	0.012	0.006	0.251	0.260
		log	0.306	0.322	**0.321**	**0.334**	0.222	0.228	-0.008	-0.010	0.006	0.001	0.210	0.219
		exp	0.306	0.322	**0.321**	**0.334**	0.237	0.244	-0.008	-0.010	0.012	0.006	0.251	0.260
		sqrt	0.306	0.322	**0.300**	**0.316**	0.237	0.244	-0.008	-0.010	0.006	0.001	0.251	0.260
	max	id	0.439	0.463	0.020	0.019	**0.272**	**0.280**	**0.091**	**0.092**	**0.254**	**0.261**	0.278	0.294
		log	0.439	0.463	0.032	0.031	**0.272**	**0.280**	**0.091**	**0.092**	**0.243**	**0.250**	0.278	0.294
		exp	0.439	0.463	0.020	0.019	**0.272**	**0.280**	**0.091**	**0.092**	**0.254**	**0.261**	0.278	0.294
		sqrt	0.439	0.463	0.020	0.019	**0.272**	**0.280**	**0.091**	**0.092**	**0.254**	**0.261**	0.278	0.294
	sum	id	**0.460**	**0.482**	0.029	0.029	0.224	0.231	-0.058	-0.062	0.132	0.135	**0.285**	**0.302**
		log	**0.460**	**0.482**	0.029	0.029	0.224	0.231	-0.058	-0.062	0.142	0.146	**0.285**	**0.302**
		exp	0.277	0.299	0.029	0.029	0.269	0.276	0.110	0.115	0.142	0.146	0.248	0.265
		sqrt	**0.460**	**0.482**	0.029	0.029	0.212	0.218	-0.048	-0.052	0.132	0.135	**0.285**	**0.302**
Pointwise Preferences	flan-t5-base		0.410	0.428	-0.120	-0.123	0.113	0.117	0.099	0.103	0.125	0.121	0.339	0.352
	flan-t5-small		0.378	0.392	-0.142	-0.144	0.167	0.170	0.130	0.144	0.271	0.275	0.287	0.310
	t5-small		0.386	0.393	0.302	0.306	0.200	0.207	0.222	0.232	0.441	0.449	0.304	0.325

Bibliography

[1] Yamen Ajjour, Henning Wachsmuth, Johannes Kiesel, Martin Potthast, Matthias Hagen, and Benno Stein. Data Acquisition for Argument Search: The args.me corpus. In Christoph Benzmüller and Heiner Stuckenschmidt, editors, *42nd German Conference on Artificial Intelligence (KI 2019)*, pages 48–59. Springer, 2019.

[2] Ron Artstein. Inter-annotator agreement. *Handbook of linguistic annotation*, pages 297–313, 2017.

[3] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. In *InCoCo@NIPS*, 2016.

[4] Alexander Bondarenko, Maik Fröbe, Meriem Beloucif, Lukas Gienapp, Yamen Ajjour, Alexander Panchenko, Chris Biemann, Benno Stein, Henning Wachsmuth, Martin Potthast, and Matthias Hagen. Overview of Touché 2020: Argument Retrieval. In Avi Arampatzis, Evangelos Kanoulas, Theodora Tsikrika, Stefanos Vrochidis, Hideo Joho, Christina Lioma, Carsten Eickhoff, Aurélie Névéol, Linda Cappellato, and Nicola Ferro, editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction. 11th International Conference of the CLEF Association (CLEF 2020)*, volume 12260 of *Lecture Notes in Computer Science*, pages 384–395. Springer, 2020.

[5] Junghoo Cho and Hector Garcia-Molina. The evolution of the web and implications for an incremental crawler. In Amr El Abbadi, Michael L. Brodie, Sharma Chakravarthy, Umeshwar Dayal, Nabil Kamel, Gunter Schlageter, and Kyu-Young Whang, editors, *VLDB 2000, Proceedings of 26th International Conference on Very Large Data Bases, September 10-14, 2000, Cairo, Egypt*, pages 200–209. Morgan Kaufmann, 2000.

[6] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *CoRR*, abs/2210.11416, 2022.

[7] Cyril W. Cleverdon. The significance of the cranfield tests on index languages. In Abraham Bookstein, Yves Chiaramella, Gerard Salton, and Vijay V. Raghavan, editors, *Proceedings of the 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Chicago, Illinois, USA, October 13-16, 1991 (Special Issue of the SIGIR Forum)*, pages 3–12. ACM, 1991.

[8] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. Overview of the TREC 2020 deep learning track. In *TREC*, 2020.

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

[10] Maik Fröbe, Christopher Akiki, Martin Potthast, and Matthias Hagen. Noise-reduction for automatically transferred relevance judgments. In Alberto Barrón-Cedeño, Giovanni Da San Martino, Mirko Degli Esposti, Fabrizio Sebastiani, Craig Macdonald, Gabriella Pasi, Allan Hanbury, Martin Potthast, Guglielmo Faggioli, and Nicola Ferro, editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 13th International Conference of the CLEF Association, CLEF 2022, Bologna, Italy, September 5-8, 2022, Proceedings*, volume 13390 of *Lecture Notes in Computer Science*, pages 48–61. Springer, 2022.

[11] Maik Fröbe, Janek Bevendorff, Lukas Gienapp, Michael Völske, Benno Stein, Martin Potthast, and Matthias Hagen. Copycat: Near-duplicates within and between the clueweb and the common crawl. In Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai, editors, *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 2398–2404. ACM, 2021.

[12] Maik Fröbe, Jan Heinrich Reimer, Sean MacAvaney, Niklas Deckers, Janek Bevendorff, Benno Stein, Matthias Hagen, and Martin Potthast.

The information retrieval experiment platform. In Michael Leyer and Johannes Wichmann, editors, *Lernen, Wissen, Daten, Analysen (LWDA) Conference Proceedings, Marburg, Germany, October 9-11, 2023*, volume 3630 of *CEUR Workshop Proceedings*, pages 175–178. CEUR-WS.org, 2023.

[13] Maik Fröbe, Matti Wiegmann, Nikolay Kolyada, Bastian Grahm, Theresa Elstner, Frank Loebe, Matthias Hagen, Benno Stein, and Martin Potthast. Continuous integration for reproducible shared tasks with tira.io. In Jaap Kamps, Lorraine Goeuriot, Fabio Crestani, Maria Maistro, Hideo Joho, Brian Davis, Cathal Gurrin, Udo Kruschwitz, and Annalina Caputo, editors, *Advances in Information Retrieval - 45th European Conference on Information Retrieval, ECIR 2023, Dublin, Ireland, April 2-6, 2023, Proceedings, Part III*, volume 13982 of *Lecture Notes in Computer Science*, pages 236–241. Springer, 2023.

[14] Lukas Gienapp, Maik Fröbe, Matthias Hagen, and Martin Potthast. The impact of negative relevance judgments on NDCG. In Mathieu d'Aquin, Stefan Dietze, Claudia Hauff, Edward Curry, and Philippe Cudré-Mauroux, editors, *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, pages 2037–2040. ACM, 2020.

[15] Lukas Gienapp, Maik Fröbe, Matthias Hagen, and Martin Potthast. Sparse pairwise re-ranking with pre-trained transformers. In Fabio Crestani, Gabriella Pasi, and Éric Gaussier, editors, *ICTIR '22: The 2022 ACM SIGIR International Conference on the Theory of Information Retrieval, Madrid, Spain, July 11 - 12, 2022*, pages 72–80. ACM, 2022.

[16] N. Jardine and Cornelis Joost van Rijsbergen. The use of hierarchic clustering in information retrieval. *Inf. Storage Retr.*, 7(5):217–240, 1971.

[17] Mosh Levy, Alon Jacoby, and Yoav Goldberg. Same task, more tokens: the impact of input length on the reasoning performance of large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 15339–15353. Association for Computational Linguistics, 2024.

[18] Sean MacAvaney and Luca Soldaini. One-shot labeling for automatic relevance estimation. In Hsin-Hsi Chen, Wei-Jou (Edward) Duh, Hen-Hsen

Huang, Makoto P. Kato, Josiane Mothe, and Barbara Poblete, editors, *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pages 2230–2235. ACM, 2023.

[19] Sean MacAvaney, Nicola Tonellotto, and Craig Macdonald. Adaptive reranking with a corpus graph. In Mohammad Al Hasan and Li Xiong, editors, *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, pages 1491–1500. ACM, 2022.

[20] Sean MacAvaney, Andrew Yates, Sergey Feldman, Doug Downey, Arman Cohan, and Nazli Goharian. Simplified data wrangling with ir_datasets. In Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai, editors, *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 2429–2436. ACM, 2021.

[21] Bernard Monjardet. On the comparison of the spearman and kendall metrics between linear orders. *Discret. Math.*, 192(1-3):281–292, 1998.

[22] Paul Owoicho, Jeff Dalton, Mohammad Aliannejadi, Leif Azzopardi, Johanne R. Trippas, and Svitlana Vakulenko. TREC cast 2022: Going beyond user ask and system retrieve with initiative and response generation. In Ian Soboroff and Angela Ellis, editors, *Proceedings of the Thirty-First Text REtrieval Conference, TREC 2022, online, November 15-19, 2022*, volume 500-338 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2022.

[23] João R. M. Palotti, Harrisen Scells, and Guido Zuccon. Trectools: an open-source python library for information retrieval practitioners involved in trec-like campaigns. In Benjamin Piwowarski, Max Chevalier, Éric Gaussier, Yoelle Maarek, Jian-Yun Nie, and Falk Scholer, editors, *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, pages 1325–1328. ACM, 2019.

[24] Ronak Pradeep, Rodrigo Frassetto Nogueira, and Jimmy Lin. The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models. *CoRR*, abs/2101.05667, 2021.

[25] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Explor-

ing the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020.

[26] Mark Sanderson. Test collection based evaluation of information retrieval systems. *Found. Trends Inf. Retr.*, 4(4):247–375, 2010.

[27] Nandan Thakur, Luiz Bonifacio, Maik Fröbe, Alexander Bondarenko, Ehsan Kamalloo, Martin Potthast, Matthias Hagen, and Jimmy Lin. Systematic evaluation of neural retrieval models on the touché 2020 argument retrieval subset of BEIR. In Grace Hui Yang, Hongning Wang, Sam Han, Claudia Hauff, Guido Zuccon, and Yi Zhang, editors, *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, pages 1420–1430. ACM, 2024.

[28] Ellen Voorhees. Overview of the TREC 2004 Robust Retrieval Track. In *TREC*, 2004.

[29] Ellen M. Voorhees. NIST TREC Disks 4 and 5: Retrieval Test Collections Document Set, 1996.

[30] Ellen M. Voorhees and Donna Harman. Overview of the Seventh Text Retrieval Conference (TREC-7). In *TREC*, 1998.

[31] Ellen M. Voorhees and Donna Harman. Overview of the Eight Text Retrieval Conference (TREC-8). In *TREC*, 1999.

[32] Shengyao Zhuang, Honglei Zhuang, Bevan Koopman, and Guido Zuccon. A setwise approach for effective and highly efficient zero-shot ranking with large language models. In Grace Hui Yang, Hongning Wang, Sam Han, Claudia Hauff, Guido Zuccon, and Yi Zhang, editors, *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, pages 38–47. ACM, 2024.

Last edited: January 2024

Declaration of Academic Integrity

1. I hereby confirm that this work — or in case of group work, the contribution for which I am responsible and which I have clearly identified as such — is my own work and that I have not used any sources or resources other than those referenced.

   I take responsibility for the quality of this text and its content and have ensured that all information and arguments provided are substantiated with or supported by appropriate academic sources. I have clearly identified and fully referenced any material such as text passages, thoughts, concepts or graphics that I have directly or indirectly copied from the work of others or my own previous work. Except where stated otherwise by reference or acknowledgement, the work presented is my own in terms of copyright.

2. I understand that this declaration also applies to generative AI tools which cannot be cited (hereinafter referred to as 'generative AI').

   I understand that the use of generative AI is not permitted unless the examiner has explicitly authorized its use (Declaration of Permitted Resources). Where the use of generative AI was permitted, I confirm that I have only used it as a resource and that this work is largely my own original work. I take full responsibility for any AI-generated content I included in my work.

   Where the use of generative AI was permitted to compose this work, I have acknowledged its use in a separate appendix. This appendix includes information about which AI tool was used or a detailed description of how it was used in accordance with the requirements specified in the examiner's Declaration of Permitted Resources.

   I have read and understood the requirements contained therein and any use of generative AI in this work has been acknowledged accordingly (e.g. type, purpose and scope as well as specific instructions on how to acknowledge its use).

3. I also confirm that this work has not been previously submitted in an identical or similar form to any other examination authority in Germany or abroad, and that it has not been previously published in German or any other language.

4. I am aware that any failure to observe the aforementioned points may lead to the imposition of penalties in accordance with the relevant examination regulations. In particular, this may include that my work will be classified as deception and marked as failed. Repeated or severe attempts to deceive may also lead to a temporary or permanent exclusion from further assessments in my degree programme.

Place and date

Signature