Methods, Applications, and Directions of Learning-to-Rank in NLP Research

Justin Lee[1] **Gabriel Bernier-Colborne**[2] **Tegan Maharaj**[1] **Sowmya Vajjala**[2]

[1] University of Toronto, Canada
chunhin.lee@mail.utoronto.ca, tegan.maharaj@utoronto.ca

[2] National Research Council, Canada
{gabriel.bernier-colborne, sowmya.vajjala}@nrc-cnrc.gc.ca

Abstract

Learning-to-rank (LTR) algorithms aim to order a set of items according to some criteria. They are at the core of applications such as web search and social media recommendations, and are an area of rapidly increasing interest, with the rise of large language models (LLMs) and the widespread impact of these technologies on society. In this paper, we survey the diverse use cases of LTR methods in natural language processing (NLP) research, looking at previously under-studied aspects such as multilingualism in LTR applications and statistical significance testing for LTR problems. We also consider how large language models are changing the LTR landscape. This survey is aimed at NLP researchers and practitioners interested in understanding the formalisms and best practices regarding the application of LTR approaches in their research.

1 Introduction

Ranking, i.e., ordering according to some property, is a central problem for many natural language processing (NLP) and information retrieval (IR) tasks such as search, question answering, document summarization, and machine translation. While NLP and IR tasks overlap, generally speaking in IR ranking problems are query-based (e.g. search, QA), while this is not necessarily true for NLP tasks. **Learning-to-rank (LTR)** is the process of applying machine learning methods to the task of ranking, i.e., to *learn* how to order elements in a sample from a data distribution. This is in contrast to performing the ranking using non-learning approaches, e.g. rule-based heuristics. LTR is commonly treated as a supervised learning problem, although research on unsupervised methods and reinforcement learning for LTR also exists (Narayan et al., 2018; Stoehr et al., 2023). In this paper we focus on the formal background of LTR and the most widely-used supervised methods. We also discuss the increasing use of large language models (LLMs) for this task, and what we expect for the future of LTR in NLP and machine learning more broadly.

An NLP problem can be framed as a ranking problem when multiple candidate solutions are present and the top k options are considered to get the final solution. This general definition fits a wide number of scenarios. For example: (1) In classification, one may set $k = 1$ and choose the top-ranked result as the solution. When the number of classes is large, or in multi-label classification scenarios, a ranking would sometimes be more suitable than choosing the most likely class. (2) In machine translation the best possible translation(s) may be chosen from a list of generated translations. (3) In generating summaries for a given text, one may modularize the problem by generating summary sentences or paragraphs separately, and then ordering them.

Discussion around LTR typically focuses on IR (e.g., web search) tasks, but many other use cases exist within NLP, as these examples show. In information retrieval tasks, LTR is generally applied to **relevance ranking**, where there is a query, and a list of instances related the query which need to be retrieved and ranked. However, in LTR for many NLP tasks the query is optional, and the core problem is to learn to rank a list of instances with respect to some property of the *list items* (e.g., ranking a set of essays based on text quality), rather than the (properties of) the *query* as in relevance ranking. Further, LTR is also sometimes used as an intermediate step in several NLP tasks (e.g., in sequence tagging, to rank the possible tags for a given token).

Three book-length works on LTR exist, to our knowledge (Liu et al., 2009; Li, 2014; Lin et al., 2021). While the first two focused on the defining the problem and described commonly used methods, Lin et al. (2021) is about how recent neural network architectures can be applied for LTR. All

Findings of the Association for Computational Linguistics: NAACL 2024, pages 1900–1917

three books primarily focused on the methods themselves and not on specific use cases within NLP. Additionally, there is little discussion on evaluation and almost none on statistical significance testing for LTR in these three books. This paper addresses this gap, and provides some guidelines on:

1. common LTR methods and evaluation measures used, including recent generative large language models (Section 2)
2. LTR use cases in NLP applications (Section 3)
3. significance testing for LTR (Section 4)

and ends by drawing some conclusions on current trends and future directions (Section 5).

2 Methods in LTR

Li (2014) describes LTR as a supervised learning problem where the training data consists of a collection of queries/requests and an associated ranked list of items for each request. Formally, the task is specified as follows: let $\{q_1, q_2, ...q_m\}$ be the set of queries, and for each query q_j, there is a set of pairs $(x_1, y_1), (x_2, y_2)..., (x_n, y_n)$ where x_i and y_i refer to the ith item and its corresponding relevance label respectively, and items can be documents/words/sentences/paragraphs. In an IR task, relevance labels signify the relevance relation between a query-item pairing. In NLP, the functionality of relevance labels can be replaced with any type of label suitable for an NLP task (i.e classes in multi-label classification, indicator labels for candidates in text generation, outputs of a machine translation system, etc.). Thus, any NLP problem can be converted into an LTR problem without altering the nature of the original problem/datasets/evaluation measures.

The modeling goal in LTR is to learn a function f that can produce scores for an optimal ranking. We note q may be null/empty, producing the queryless ranking problems we will discuss in Section 3.2. LTR methods can take on a variety of forms with solutions that directly or indirectly optimize for a ranking metric. They can be categorized into three groups as pointwise, pairwise, and listwise methods, which differ based on the choice of loss function and input representation.

For a given task to be framed as a ranking problem, the data is often partitioned into groupings, usually corresponding to a query. So, each grouping contains a query and a set of items to rank related to the query. In the absence of an explicit query, groupings still exist and are used during training and evaluation to designate appropriate aggregations for metric calculation (e.g. averaging over groupings) and for splitting into appropriate train-test-validation sets. In this section, we will discuss some of the commonly used loss functions, supervised learning methods, and evaluation measures.

2.1 Loss Functions

A **pointwise** LTR method aims to learn a function f with parameters θ and a loss function L such that $f(q, x_i, \theta) = \hat{y}_i$ and $L(y_i, \hat{y}_i)$ is minimized. The loss function L can take the form of the mean squared error (if the relevancy labels are continuous) or a cross entropy loss (if the relevancy labels are categorical). Unlike other LTR methods, the pointwise methods do not directly optimize for ranking metrics, and the learning task can be framed as a regression or classification problem. However, the predictive scores learned from these models are then used to order an input list of items (rather than a direct classification or prediction task). These approaches are the most simple forms of LTR, but are often useful as preliminary baseline scores for more complex methods.

A **pairwise** LTR method aims to learn the query-item relevance between pairs of items. For $(x_i, y_i), (x_j, y_j)$, a pair of resulting items for a query q, a pairwise training label y'_{ij} can be formed by taking the difference of the relevance labels y_i and y_j, i.e. $y'_{ij} = y_i - y_j$ where y'_{ij} would be positive if $y_i > y_j$ and negative if $y_i < y_j$, and this can be treated as a binary label. To form the input representation, features are constructed by applying an operation (e.g. difference) to the features of both data points in a given pair (Joachims, 2002). Depending on the size of groupings, computational complexity may be a challenge if all pairwise permutations are to be constructed. Tymoshenko et al. (2017) presents an example of an alternating pairwise algorithm used to construct training examples, while Lee and Vajjala (2022) use a sampling method that anchors the lowest and highest ranked examples and uniformly samples a data point in-between these examples. The pairwise LTR function f then takes the form of $f(q, x_i, x_j, \theta) = \hat{y}'_{ij}$. A loss function $L(y'_{ij}, \hat{y}'_{ij})$ is minimized, which is usually the cross-entropy loss.

A **listwise** LTR method aims to learn a function to estimate a full list of scores to rank the item list.

It takes the form $f(q, [x_1, .., x_n], \theta) = [\hat{s}_1, ..., \hat{s}_n]$. A ranking is then obtained by sorting $[\hat{s}_1, ..., \hat{s}_n]$ in descending order. In past work, listwise methods made use of the permutation probability and the top-one probability as the learning objective (Cao et al., 2007).

2.2 Ranking Models

While pointwise methods are often covered in papers referring to LTR models, this section will mainly focus on models that implement pairwise and listwise objectives due to their popularity in applied NLP.

Pairwise Models: SVMrank (Joachims, 2002) frames the pairwise ranking objective within the SVM algorithm, and has been a popular choice in NLP. After the feature distances and indicator binary labels are applied to pairs of ranking data, the problem is treated as an SVM classification problem. Models with outputs that are differentiable functions of parameters are also very popular: RankNet (Burges et al., 2005), LambdaRank (Burges et al., 2006), and LambdaMART (Burges, 2010) use gradient descent to update a pairwise model. RankNet explicitly defines a cost function to update via gradient descent, while LambdaRank bypasses this in favor of directly defining a gradient function that can optimize for a specific metric. LambdaMART implements the LambdaRank objective with boosted regression trees.

Pairwise ranking objectives have also been optimized by modern neural network architectures for NLP tasks. Lee and Vajjala (2022) fine-tuned a transformer (Vaswani et al., 2017) model on the pairwise ranking objective for automatic readability assessment, while dos Santos et al. (2015) used a convolutional neural network (CNN) to learn the relationship between nominals in a sentence. However, SVMrank remains one of the most popular non-neural methods for pairwise ranking in NLP, and is often listed as a competitive baseline.

Listwise ListNet (Cao et al., 2007) is a probabilistic ranking model where the probability of a list item being ranked in the first position given the all other items in a list, is predicted per item in the list. ListMLE (Xia et al., 2008) builds on ListNet by proposing an alternative loss function that has a number of desirable properties (i.e order sensitive, good approximation of a binary loss on permutations, continuous and differentiable). As with the pairwise methods, transformer models have also been used to optimize for listwise objectives. ListBERT (Kumar and Sarkar, 2022) finetunes a RoBERTa model with several listwise losses for ranking e-commerce products, whereas Yan et al. (2020) propose a listwise ranker based on a recurrent neural network (RNN) auto-encoder for ranking biomedical question-answer pairs.

2.3 Contrastive Learning

Chopra et al. (2005) describe a supervised or self-supervised learning objective where a loss function is designed to enforce similar representations for data of the same category, and dissimilar representations for data of different categories (Jaiswal et al., 2021). First introduced in computer vision literature, this type of learning objective and relevant loss functions have been popular in NLP under the name “contrastive learning”. This has been explored in NLP for some ranking tasks in recent years (Reimers and Gurevych, 2019; Briakou and Carpuat, 2020; Gao et al., 2021; Min et al., 2022; Chernyavskiy et al., 2022; Liu et al., 2023; Rau and Kamps, 2022).

2.4 Ranking with Generative Models

Generative sequence-to-sequence models have also been used to tackle ranking problems in the recent past. Unlike the BERT-based methods that optimize pairwise or listwise losses, generative models use a prompt-based approach, which outputs tokens rather than numerical scores, and ranking problems are treated accordingly. For example, Nogueira et al. (2020) used a pre-trained T5 model (Raffel et al., 2020), an encoder-decoder model, to rank documents by specifying an input template with slots for “Query”, “Document”, and “Relevant” and the relevance score is obtained by applying a softmax function on the logits for the output tokens “true” and “false”, which are analogous to binary relevance labels. RankT5 (Zhuang et al., 2022) fine-tunes the T5 model to extend to both pairwise and listwise ranking losses.

An increasing body of recent research explores using decoder-only LLMs as (re)rankers. Ji et al. (2023) investigated ChatGPT’s ability to rank the outputs of various models on a diverse set of tasks including NLP tasks and open-ended generation tasks. Most work on LLM-based (re)ranking evaluates their performance on query-focused, IR tasks. This is typically done in two stages: retrieval and reranking. Given the query, an set of candidates is first retrieved from the large pool of passages or documents, using either an LLM for dense retrieval or a more efficient search method, e.g. BM25 (Robertson and Zaragoza, 2009); then these candidates are reranked using the LLM for improved ranking accuracy. This can be done with or without fine-tuning. For example, Ma et al. (2023a) fine-tuned an LLM both for both dense retrieval and for pointwise reranking, and another pointwise reranking approach based on instruction distillation was proposed by Sun et al. (2023a).

Other work has shown that LLMs are effective rerankers in zero-shot settings. Liang et al. (2023) used zero-shot prompting for pointwise ranking: they prompt the model to predict whether document a relevant is relevant to query q, and score by the probability of the answer being "Yes". Qin et al. (2023) used a pairwise approach: they prompt the model to predict whether document a is more relevant than document b to query q. Listwise reranking approaches take the candidate documents and generate a reordered list of document identifiers (Ma et al., 2023b; Sun et al., 2023b; Pradeep et al., 2023a,b; Tang et al., 2023). Zhuang et al. (2023) tested LLMs as query likelihood models in both zero-shot and few-shot settings. Experiments have shown listwise approaches to be more effective than pointwise or pairwise (Ma et al., 2023b; Pradeep et al., 2023a), and the increasing context window sizes of LLMs make them increasingly attractive. For more information on the usage of generative language models for search and recommendation tasks, we refer the reader to the surveys by Zhu et al. (2024) and Wu et al. (2023).

Ranking also plays a role in an increasingly popular workflow called retrieval-augmented generation (RAG). Here, given a query, a small subset of relevant documents is retrieved and ranked, and an LLM then generates the output using the retrieved documents as additional context (Gao et al., 2023).

2.5 Software Tools

Ranklib[1] has implementations for a variety of LTR algorithms and XGBoost[2], a popular library for gradient boosted models, contains an implementation of LambdaRank. Tensorflow Ranking[3] is an open-source library for developing neural ranking models and AllRank is a similar open-source library for PyTorch [4]. Recent work on LlamaIndex[5] and LangChain[6] provide an interface for connecting LLMs with indexed, textual data to be ranked.

2.6 Evaluation

The choice of evaluation measure when using LTR methods in NLP applications primarily depends on whether the task is that of relevance ranking of items for a given query or ranking a full list of items without such a query. Some commonly used evaluation measures are listed below grouped into two categories accordingly.

Evaluating ranking for a given query : Normalized Discounted Cumulative Gain (**NDCG**) and Discounted Cumulative Gain (**DCG**) (Järvelin and Kekäläinen, 2017) are measures of the goodness of a ranked list in terms of relevance, and are commonly used in retrieval tasks. **P@k, R@k, F1@K** i.e., Precision/recall/F1 score with a cut-off at kth position (typically, $k = 5$ or 10) are also used in relevance ranking tasks (e.g., ranking of keyphrases). Mean Reciprocal Rank (**MRR**) and Mean Average Precision (**MAP**) are measures commonly reported in question-answering tasks, where there may typically be a single best answer. Reciprocal rank is the inverse of the rank of the best answer and MAP is the mean of the average precision, i.e., the area under the precision recall curve. Both these are not used in situations where the ranking of the entire list is relevant, and are not commonly reported in NLP use cases of LTR.

Evaluating ranking without an explicit query: When there are two ranked lists, one from a ranking model and one ground truth ranking, **Kendall's Tau** (τ) and **Spearman's rank correlation** (ρ) are used to compare the two ranked lists. Pearson correlation is also sometimes used in such cases. **Ranking Accuracy/Perfect Match Ratio**, which is the proportion of data instances where the ranking order from the model exactly matches the reference order, is also a commonly measure. One major difference among these metrics is their approach towards handling ties. While ranking accuracy does not handle ties, τ penalizes ties in ground-truth and predictions, and ρ calculates the average rank of ties. Some recent research (Lee and Vajjala,

[1] https://sourceforge.net/p/lemur/wiki/RankLib/
[2] https://xgboost.readthedocs.io/en/stable/
[3] https://www.tensorflow.org/ranking
[4] https://github.com/allegro/allRank
[5] https://gpt-index.readthedocs.io/en/latest/index.html
[6] https://python.langchain.com/en/latest/index.html

2022) recommends reporting multiple evaluation measures due to these differences.

Software libraries such as SciPy (Virtanen et al., 2020), scikit-learn (Pedregosa et al., 2011), ranx[7], evaluate[8] and TREC-eval[9] have implementations of most of these metrics. While some research explored new evaluation measures for specific ranking tasks such as information ordering (Lapata, 2006; Madnani et al., 2007) and temporal ordering of events (Jeblee and Hirst, 2018), or improving existing measures (Katerenchuk and Rosenberg, 2016), such measures were not widely adopted into mainstream LTR research in NLP.

The discussion in this section focused on the general methods for LTR including the nature of data, modeling techniques, and evaluation measures, and how an NLP problem and corresponding datasets can be viewed through an LTR lens. In the next section, we look into how LTR methods are used across various NLP applications in practice.

3 Overview of LTR Applications in NLP

In previous surveys, ranking approaches have been separated as **ranking creation** (ordering according to a criteria, with or without a query) vs. **ranking aggregation** (combining previously-computed rankings) (Li, 2014), or **re-ranking** (of a previously computed ranking) vs. **direct ranking**, sometimes called dense ranking or dense retrieval (Lin et al., 2021). In this paper, we distinguish the applications based on whether the ranking is done when there is a query/reference and a set of items to be ranked in terms of relevance to the query, i.e. **ranking with a query** (Section 3.1) vs. when there is no explicit query, only a list of items to be ranked , i.e. **ranking without a query**(Section 3.2)[10]. Aligning with the growing efforts in the NLP community on studying and expanding multilingual support for NLP applications, we note the multilingual coverage of LTR use cases within NLP where possible.

3.1 Ranking with a Query

Question answering, which involves tasks such as selection and ranking of relevant passages for a given question, extracting answer spans from each passage or choosing from a multiple-choice setup is a classic example of ranking with a query. A related task is community question answering, where a similarity-based ranking of other questions that are close to the user's query is performed. These are the commonly seen use cases of LTR in NLP research, and a range of methods from traditional ranking approaches to tree kernels and convolutional neural networks were explored (e.g., Belinkov et al., 2015; Louis and Lapata, 2015; Malhas et al., 2016; Tymoshenko et al., 2017; Do et al., 2017; Pirtoaca et al., 2019; D'Souza et al., 2019; Yan et al., 2020; Zhang et al., 2023). Except Belinkov et al. (2015) and Malhas et al. (2016) who used Arabic datasets, all others cited worked only with English datasets.

LTR as the primary task LTR methods are applied in several NLP tasks that involve the creation of a ranked list from the given set of items. Ranking texts by relevance to a given query (Severyn and Moschitti, 2015), query-focused single and multi-document summarization (Jin et al., 2010; Yin et al., 2012; Cao et al., 2016; Lu et al., 2016; Liu and Xu, 2023), re-ranking of n-best outputs in machine translation (Shen et al., 2004; Li et al., 2013; Niehues et al., 2015; Li and Wang, 2018; Lee et al., 2021) and optical character recognition (Tomeh et al., 2013) are examples of such tasks that have some form of ranking problem in their pipeline. There are several other NLP applications of this kind, such as choosing best headlines for a given article (Kourogi et al., 2015; Higurashi et al., 2018), ranking tweets by their credibility with respect to an event (Gupta and Kumaraguru, 2012), ranking relevant reviews for medical products in e-commerce applications (Uppal et al., 2019), ranking reader emotions in a given document (Lin and Chen, 2008), and differential diagnosis, using LTR to find the most probable diseases given a clinical description text (Amiri et al., 2021). LTR methods are also used on sub-sentence level for tasks such as ranking of potential words/phrases for lexical substitutions (Szarvas et al., 2013; Liang et al., 2018; Paetzold and Specia, 2017) or ranking keyphrases (Eichler and Neumann, 2010)

Amongst these, excluding papers working with machine translation datasets, only three papers (Tomeh et al., 2013; Kourogi et al., 2015; Higurashi et al., 2018) reported experiments with non-English (Arabic and Japanese) datasets. Pairwise ranking methods are more commonly used, although some reported comparisons with listwise methods, and

[7] https://github.com/AmenRa/ranx

[8] https://huggingface.co/evaluate

[9] https://github.com/usnistgov/trec_eval

[10] Our process for selecting the relevant papers is explained in Appendix A. See also Appendix B for more information.

showed either comparable or slightly better results over pairwise methods (Lu et al., 2016; Jin et al., 2010; Yin et al., 2012; Szarvas et al., 2013).

LTR as an intermediate step: The tasks mentioned so far have a ranking/ordering problem specified in their definition. However, LTR methods have also been used as an intermediate step for other standard NLP tasks that are not particularly specified as a ranking task, to choose the final prediction for the NLP model, among the possible options. For example, Ji et al. (2006) and Darwish et al. (2017b) use LTR for sequence tagging problems, to rank the possible tags for a given word. Entity linking (Zheng et al., 2010; Chen and Ji, 2011; McNamee et al., 2011), morphological analysis (Darwish and Mubarak, 2016; Darwish et al., 2017a), coreference resolution (Irwin et al., 2011; Tran et al., 2011), referring expression generation (Zarrieß and Kuhn, 2013), surface realizations in text generation (Zarrieß et al., 2012; Mazzei and Basile, 2019), and slot ranking in dialog systems (Wang et al., 2022) are other examples of this kind, where LTR methods were used in the pipeline of some classic NLP tasks.

A few other examples include: disease normalization i.e., determining which diseases are mentioned in the text (Leaman et al., 2013), identifying phrasal verbs (Pichotta and DeNero, 2013), short answer scoring (Mohler et al., 2011), choosing the target languages in cross-lingual transfer (Lin et al., 2019), and ranking of labels in multi-label text classification (Azarbonyad et al., 2021), knowledge graphs (Gao et al., 2022), language modeling (Frydenlund et al., 2022), fact-checking (Fajcik et al., 2023), and ensembling of LLM outputs (Jiang et al., 2023).

While pairwise methods (especially SVMrank) dominate here too, listwise approaches were found to be useful for some of the tasks (e.g., coreference resolution (Tran et al., 2011), surface realization, the task of generating linear form of a text given a syntactic representation (Mazzei and Basile, 2019), multilabel classification (Azarbonyad et al., 2021)). In terms of non-English datasets, LTR was used with Arabic (Darwish and Mubarak, 2016; Darwish et al., 2017b,a), Spanish and Catalan (Tran et al., 2011), German (Zarrieß et al., 2012; Zarrieß and Kuhn, 2013), and French (Mazzei and Basile, 2019) and Chinese (Mazzei and Basile, 2019; Jiang et al., 2023) across a range of tasks. Clearly there is more language diversity in this set of tasks compared to others that used LTR in NLP so far.

3.2 Ranking without a Query

In NLP, it is common to see problems that seek a ranking of items without a specific query. Information ordering tasks, where the goal is to rank a given set of items based on a criteria (e.g., coherence, polarity, formality, readability, etc.) are examples of tasks of this kind. LTR has also been studied as an alternative to classification and regression in tasks such as readability assessment and essay scoring where there is no associated query. While the ranking methods used themselves are not different in such cases, the evaluation measures used are often different from the ones used where there is a reference/query (see Section 2.6 for a discussion).

Text summarization without a reference query is an example where LTR methods have been used to rank sentences (Narayan et al., 2018). Ordering the sentences in a paragraph (Kumar et al., 2020), and temporal ordering of events in clinical notes (Jeblee and Hirst, 2018) are other examples.

Readability assessment is the problem of determining the readability of a text. In this task, the input is comprised of lists of texts to be ranked by readability, and the outputs are the same lists of texts, sorted by readability. Pairwise ranking has been well studied for this task (Pitler and Nenkova, 2008; Tanaka-Ishii et al., 2010; Ma et al., 2012; Vajjala and Meurers, 2016; Liu et al., 2018; Lee and Vajjala, 2022) and recent research (Lee and Vajjala, 2022) showed that a pairwise ranking based approach performed better in cross-domain and cross-lingual transfer scenarios for this task. Of these, while Tanaka-Ishii et al. (2010) reported results on English and Japanese datasets, Lee and Vajjala (2022) employed English, French and Spanish datasets.

Essay scoring is the task of evaluating student/learner essays and assignments automatically. While this is generally modeled as a classification/regression problem, a popular approach is to order a collection of student writings instead of grading them separately. Yannakoudakis et al. (2011), Kuzi et al. (2019) and Yang et al. (2020) demonstrated the usefulness of ranking methods for English essay scoring.

Ranking words/phrases in problems where the input is a set of words, and the output is a set of scores which can then be ranked, such as sentiment intensity ranking (Wang et al., 2016), polarity and formality ranking (Brooke and Hirst, 2014) can also be considered as examples of tasks without a query. Wang et al. (2017) discuss ranking approaches for measuring semantic coherence between pairs of texts. Of these, only MacLaughlin and Smith (2021) mentions working with non-English (Latin) data along with English. Ranking speakers in terms of their relative power in political debates (Prabhakaran et al., 2013, 2014), documents for plagiarism detection (Chong and Specia, 2012) and passages in a document in terms of their quotability (MacLaughlin and Smith, 2021), and ranking different versions of a claim for quality Skitalinskaya et al. (2021) are some uncommon examples.

While this list is not exhaustive, these examples demonstrate the diverse usage of LTR methods for various NLP tasks, and how many of the use cases are different from the traditional IR task of relevance ranking of a set of items in response to a query. This diversity also resulted in the use of many different, task-specific and language-specific datasets while using LTR in NLP. Our main observations are summarized as follows:

1. Although pointwise/pairwise/listwise ranking approaches have all been explored for various NLP tasks, pairwise ranking is the most commonly used approach. While we did not find any noticeable trend in the choice among these approaches, it has to be noted that pairwise methods are relatively easier to implement and even standard binary classification techniques can be used to learn to compare pairs, whereas listwise methods require more careful consideration, and are computationally more intensive, which could explain the preference for pairwise LTR methods in NLP.

2. In terms of multilinguality, only about 22% of the papers listed in this section explored non-English datasets (16/73), with Arabic used across five tasks.

Note that LTR approaches are not necessarily the best-performing solution for some of the tasks and traditional classification or regression approaches may be better solutions, based on the nature of the task. Our aim in this section is only to provide an overview of where (and how) LTR methods are adapted for various NLP tasks, not to assess whether they are the best-performing approach for a given task.

4 Significance Testing

As this survey aims to guide NLP researchers and practitioners, we consider it important to discuss not only how to implement and evaluate ranking, but also how to reliably compare different methods. Therefore, in this section, we present an overview of significance testing methods, before analyzing the actual usage of such methods in the papers we surveyed and providing recommendations.

4.1 Methods

The goal of significance testing is to determine the probability that the difference in score between two algorithms, termed the "test statistic", is due to chance. If the difference is indeed due to chance, the true expected value of the test statistic is 0 – this is termed the "null hypothesis". More formally, the test statistic δ is defined as the absolute difference in scores between two models on some test set D, i.e. $\delta = |m_1(D) - m_2(D)|$. The null hypothesis is that the true value of $\delta = 0$. The probability of obtaining a δ greater than or equal to the observed value, assuming the null hypothesis, is called the "p-value". If p is smaller than some pre-determined significance level (usually 0.05 or 0.01), the null hypothesis is rejected, and the difference is considered significant.

In IR research, significance testing has become the norm in shared tasks such as those at TREC (Voorhees and Harman, 2005), and some studies compared the suitability and reliability of statistical significance tests on common evaluation measures (Sanderson and Zobel, 2005; Parapar et al., 2021). Regarding NLP specifically, one useful reference on hypothesis testing is the textbook by Dror et al. (2020), as well as the papers on which it is based (Dror et al., 2017, 2018, 2019). The book includes a survey of the most relevant significance tests for common NLP tasks, matching tasks and their evaluation measures with the most appropriate test. In NLP settings, significance tests are usually paired, which means that they compare the results of two algorithms on every example in the test set, and then provide an aggregate p-value. There are several types of tests.

Parametric tests make assumptions about the distribution of the test statistic under the null hypothesis (typically normality). They are less likely than non-parametric tests to accept the null hypothesis when it should be rejected, but if the distribution is unknown, non-parametric tests should be used instead. The paired student's t-test (Fisher, 1935) is the most popular parametric test in NLP.

Non-parametric tests can be grouped into sampling-free and sampling-based methods. Sampling-free tests include several variations of the sign test including Wilcoxon's signed rank test (Wilcoxon, 1945). This test ranks the test cases by the difference between the two scores (large to small), then sums the signed ranks of this ordered list of test cases. This test "is actually applicable for most NLP setups" (Dror et al., 2018).

Sampling-based tests include the Fisher-Pitman permutation test (Pitman, 1937; Fisher, 1935; Noreen, 1989) and the bootstrap test (El Barmi and McKeague, 2013). These tests are more robust because they consider the actual values of the test statistic, not just the signed ranks; on the other hand, they are more computationally expensive. The permutation test checks how often δ is greater than the observed value if we randomly swap the scores of the two systems and consider all permutations (or some random sample if that is unfeasible). The bootstrap test is similar, but we sample test cases with replacement from the actual test set rather than randomly swapping outputs.

Dror et al. (2018) propose a simple decision tree to select the appropriate test: if the distribution of the test statistic is known (or can be shown to be normal or similar to some reference distribution), we should prefer a parametric test. Otherwise, we should prefer sampling-based methods as long as the test set is not too small (because of the sampling error) or too large (for computational feasibility), and sampling-free methods otherwise.

Other approaches Evert (2004) presents a model-based approach which he applies to the task of collocation extraction, a query-less task. This method assumes that precision scores are the result of a random experiment and follow a binomial distribution. The null hypothesis, i.e., that the distribution means are the same, is tested using Fisher's exact test. Similarly, Goutte and Gaussier (2005) compute a distribution for the evaluation metric (focusing on precision, recall, and F-score), then sample from the distributions of two systems to test for a significant difference. Riezler and Hagmann (2021) proposed another model-based method, which is applicable to a wide range of evaluation measures, and can handle hyperparameter variation and multiple test sets.

Bayesian approaches (Gelman et al., 2020) have also been used for hypothesis testing in NLP. Sadeqi Azer et al. (2020) compare various hypothesis testing approaches, including Bayesian ones, on the question answering task, and provide guidelines for selecting the best approach based on the kinds of hypotheses they support. Whereas frequentist approaches produce a single point estimate of the p-value, Bayesian methods produce a probability distribution for the test statistic. The Bayesian analog of confidence intervals and p-values can then be computed. Bayesian approaches are easier to interpret and more robust to the size of the test set. So far, Bayesian hypothesis testing has been focused on classification tasks (Carrasco et al., 2020), but there exists a Bayesian version of Wilcoxon's signed rank test (Benavoli et al., 2014), which is applicable to many different tasks and evaluation measures.

There are still open issues regarding significance tests. They generally assume that test cases are independent and identically distributed (IID), but this is rarely the case in NLP data, as test sets can contain sentences from the same document, author, source, etc. (Dror et al., 2018) How to handle evaluation scores based on cross-validation is another open issue (Raschka, 2018). Note that some of the resources covered in this section provide a toolkit or experimental scripts for significance testing (Raschka, 2018; Dror et al., 2020; Carrasco et al., 2020; Sadeqi Azer et al., 2020), and that significance tests are implemented in some libraries for scientific computing, such as SciPy (Virtanen et al., 2020).

4.2 Actual Usage of Significance Testing

To assess actual usage of significance testing in this area, we inspected all the works cited in this survey for mentions of significance testing. We focused on papers reporting experiments that compare different algorithms, and excluded survey papers, papers that are specifically about significance testing itself, IR evaluation practices, toolkits, etc. This leaves a total of 108 papers.

The most frequently used test was the paired t-test, which was applied in 15 papers (see Appendix C for details) to a wide variety of metrics including precision, recall, F-score, MAP, generalized average precision, P@K, MRR@k, NDCG, correlation measures, perplexity and metrics used for coreference resolution. The second-most frequent was Wilcoxon's test, used in six papers. Various tests were used once in four different papers.

An un(der)specified test was used in a further 11 papers. Finally, Evert (2004) and Goutte and Gaussier (2005) both proposed novel tests which they then applied to NLP ranking tasks. The latter was also used by Fajcik et al. (2023).

This leaves 69 of 108 papers (64%) that do not report statistical significance. Note that in a few papers, it was difficult to determine from the text if and how significance testing was performed (due to vague usage of the term "significant"), so the statistics we provide are approximate. At any rate, there is still a tendency not to report statistical significance in this line of work. However, as some have noted, bringing about statistical reforms in a field may take a lot of effort and time (Sakai, 2014).

In summary, we recommend the following:

- To compare systems reliably, significance testing should be required. This could include adding this to so-called "responsible NLP checklists" for publication. Common tests are easy to carry out, thanks to toolkits and libraries that implement them.
- Additional statistical measures should also be considered (Sakai, 2014; Fuhr, 2017), such as confidence intervals (to assess the *reliability* of each score) and effect sizes (to quantify the actual gain provided by one algorithm over another).
- Various tests are available and there is a lot of variability in actual usage, with the t-test currently being the most common. Dror et al. (2018) provide useful guidance on choosing an appropriate test, but neglects some approaches, e.g. Bayesian methods.
- We would implore researchers to avoid describing gains as being "significant" when no appropriate test has been applied. Also, when discussing significance, it is important to remember that statistical significance does not necessarily entail *practical* significance (Hull, 1993, inter alia).

5 Conclusions and Future Directions

This survey shows a snapshot of LTR methods and current practices in the use of LTR in NLP, and provides guidance and resource pointers for significance testing, an under-practiced element of evaluation. Our key insights so far are summarized below:

1. LTR is applied in a diverse range of tasks in NLP beyond the traditional information retrieval task, resulting in the usage of many different kinds of task-specific datasets and evaluation measures. Most of this research is dominated by English datasets, though, with 22% of papers reporting on non-English datasets.

2. Pairwise approaches are more commonly adopted in NLP literature than listwise approaches, with an increasing interest in ranking with generative models and the use of LLMs in zero-shot settings in recent times.

3. Significance testing is not a common practice in this field and some papers report with unspecified tests. We summarized the available literature on the appropriate tests for LTR tasks and evaluation measures, offering recommendations for doing significance testing for LTR in NLP, and found that most (approx. 64%) of papers surveyed reported no significance testing.

Future Directions: There has been growing interest in using LLMs as zero-shot rankers (see Section 2.4 for a discussion), following the current trend of using large language models as natural language APIs. This strand of research has been primarily focused on (re-)ranking for information retrieval and question answering. We expect this trend to continue, and hyphothesize that new use cases within NLP could emerge for ranking and learning-to-rank.

The information retrieval community has been working on increasing the diversity of rankings (Radlinski et al., 2008; Haldar et al., 2022), which could be relevant for NLP problems that rely on sampling techniques and diverse text generation (e.g., machine translation, keyphrase generation). There is also emerging research on how neural ranking models can benefit from traditional IR or LTR methods (Zhang et al., 2021; Saha et al., 2023), and equivalent ideas on the relevance of traditional NLP to ranking may emerge. We hope to see more research applying significance testing across LTR in NLP and more multilingual LTR use cases in future.

6 Limitations

While LTR methods have been effective in NLP, the IR community has traditionally utilized LTR methods to a greater extent. Since work in IR can include the search and retrieval of textual data, there is not a clear boundary between LTR methods for IR and LTR methods specific to NLP use cases. For this study, we have chosen to cover LTR applications in tasks that are NLP-specific, opting against more general IR-centric LTR approaches that may operate on textual data as a medium. However with the current trend of retrieval-augmented generation with LLMs, we anticipate that these boundaries will be blurred even more in the near future. Additionally, we focused mainly on supervised LTR approaches in this paper, which overlooks other applications of LTR which follow unsupervised methods or use reinforcement learning and other approaches for learning to rank. We also considered non-NLP tasks to be out-of-scope of this work, however LTR may of course be applied to other domains – other discrete domains may find much of the work described here transferrable, but there are nuances and tricks for LTR in continuous domains which are not covered by this survey. Finally, it should be noted that our approach to selecting the papers (described in Appendix A) poses limitations on the coverage of this survey.

Acknowledgements

We thank Rebecca Knowles, Taraka Rama and James Wang, and all the anonymous reviewers for their useful feedback. Gabriel Bernier-Colborne and Sowmya Vajjala contributed to the manuscript on behalf of the National Research Council of Canada, thereby establishing a copyright belonging to the Crown in Right of Canada, that is, to the Government of Canada.

References

Hadi Amiri, Mitra Mohtarami, and Isaac Kohane. 2021. Attentive multiview text representation for differential diagnosis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 1012–1019, Online. Association for Computational Linguistics.

Hosein Azarbonyad, Mostafa Dehghani, Maarten Marx, and Jaap Kamps. 2021. Learning to rank for multi-label text classification: combining different sources of information. *Natural Language Engineering*, 27(1):89–111.

Yonatan Belinkov, Mitra Mohtarami, Scott Cyphers, and James Glass. 2015. VectorSLU: A continuous word vector approach to answer selection in community question answering systems. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 282–287, Denver, Colorado. Association for Computational Linguistics.

A. Benavoli, F. Mangili, G. Corani, M. Zaffalon, and F. Ruggeri. 2014. A Bayesian Wilcoxon signed-rank test based on the Dirichlet process. *Proceedings of the 32th International Conference on Machine Learning ICML*, 2014:1026 – 1034. Cited by: 31.

Eleftheria Briakou and Marine Carpuat. 2020. Detecting Fine-Grained Cross-Lingual Semantic Divergences without Supervision by Learning to Rank. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1563–1580, Online. Association for Computational Linguistics.

Julian Brooke and Graeme Hirst. 2014. Supervised ranking of co-occurrence profiles for acquisition of continuous lexical attributes. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2172–2183, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96.

Christopher Burges, Robert Ragno, and Quoc Le. 2006. Learning to rank with nonsmooth cost functions. *Advances in neural information processing systems*, 19.

Christopher JC Burges. 2010. From RankNet to LambdaRank to LambdaMART: An overview. *Learning*, 11(23-581):81.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136.

Ziqiang Cao, Wenjie Li, and Dapeng Wu. 2016. PolyU at CL-SciSumm 2016. In *Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL)*, pages 132–138.

J. Carrasco, S. García, M.M. Rueda, S. Das, and F. Herrera. 2020. Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review. *Swarm and Evolutionary Computation*, 54:100665.

Zheng Chen and Heng Ji. 2011. Collaborative ranking: A case study on entity linking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 771–781, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Anton Chernyavskiy, Dmitry Ilvovsky, Pavel Kalinin, and Preslav Nakov. 2022. Batch-softmax contrastive loss for pairwise sentence scoring tasks. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 116–126, Seattle, United States. Association for Computational Linguistics.

Miranda Chong and Lucia Specia. 2012. Linguistic and statistical traits characterising plagiarism. In *Proceedings of COLING 2012: Posters*, pages 195–204.

Sumit Chopra, Raia Hadsell, and Yann LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE.

Kareem Darwish and Hamdy Mubarak. 2016. Farasa: A new fast and accurate Arabic word segmenter. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1070–1074, Portorož, Slovenia. European Language Resources Association (ELRA).

Kareem Darwish, Hamdy Mubarak, and Ahmed Abdelali. 2017a. Arabic diacritization: Stats, rules, and hacks. In *Proceedings of the Third Arabic Natural Language Processing Workshop*, pages 9–17, Valencia, Spain. Association for Computational Linguistics.

Kareem Darwish, Hamdy Mubarak, Ahmed Abdelali, and Mohamed Eldesouki. 2017b. Arabic POS tagging: Don't abandon feature engineering just yet. In *Proceedings of the Third Arabic Natural Language Processing Workshop*, pages 130–137, Valencia, Spain. Association for Computational Linguistics.

Phong-Khac Do, Huy-Tien Nguyen, Chien-Xuan Tran, Minh-Tien Nguyen, and Minh-Le Nguyen. 2017. Legal question answering using ranking SVM and deep convolutional neural network. *arXiv preprint arXiv:1703.05320*.

Rotem Dror, Gili Baumer, Marina Bogomolov, and Roi Reichart. 2017. Replicability analysis for natural language processing: Testing significance with multiple datasets. *Transactions of the Association for Computational Linguistics*, 5:471–486.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392, Melbourne, Australia. Association for Computational Linguistics.

Rotem Dror, Lotem Peled-Cohen, Segev Shlomov, and Roi Reichart. 2020. Statistical significance testing for natural language processing. *Synthesis Lectures on Human Language Technologies*, 13(2):1–116.

Rotem Dror, Segev Shlomov, and Roi Reichart. 2019. Deep dominance - how to properly compare deep neural models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2773–2785, Florence, Italy. Association for Computational Linguistics.

Jennifer D'Souza, Isaiah Onando Mulang', and Sören Auer. 2019. Team SVMrank: Leveraging feature-rich support vector machines for ranking explanations to elementary science questions. In *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*, pages 90–100, Hong Kong. Association for Computational Linguistics.

Kathrin Eichler and Günter Neumann. 2010. DFKI KeyWE: Ranking keyphrases extracted from scientific articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 150–153, Uppsala, Sweden. Association for Computational Linguistics.

Hammou El Barmi and Ian W. McKeague. 2013. Empirical likelihood-based tests for stochastic ordering. *Bernoulli: Official Journal of the Bernoulli Society for Mathematical Statistics and Probability*.

Stefan Evert. 2004. Significance tests for the evaluation of ranking methods. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 945–951, Geneva, Switzerland. COLING.

Martin Fajcik, Petr Motlicek, and Pavel Smrz. 2023. Claim-dissector: An interpretable fact-checking system with joint re-ranking and veracity prediction. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10184–10205, Toronto, Canada. Association for Computational Linguistics.

Ronald Aylmer Fisher. 1935. *The Design of Experiments*. Oliver and Boyd.

Arvid Frydenlund, Gagandeep Singh, and Frank Rudzicz. 2022. Language modelling via learning to rank. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10636–10644.

Norbert Fuhr. 2017. Some common mistakes in IR evaluation, and how they can be avoided. *SIGIR Forum*, 51(3):32–41.

Hanning Gao, Lingfei Wu, Po Hu, Zhihua Wei, Fangli Xu, and Bo Long. 2022. Graph-augmented learning to rank for querying large-scale knowledge graph. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 82–92, Online only. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. 2020. *Bayesian Data Analysis, Third Edition*. Online.

Cyril Goutte and Eric Gaussier. 2005. A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In *Proceedings of the European Colloquium on IR Resarch (ECIR'05)*, pages 345–359.

Aditi Gupta and Ponnurangam Kumaraguru. 2012. Credibility ranking of tweets during high impact events. In *Proceedings of the 1st workshop on privacy and security in online social media*, pages 2–8.

Malay Haldar, Mustafa Abdool, Liwei He, Dillon Davis, Huiji Gao, and Sanjeev Katariya. 2022. Learning to rank diversely at Airbnb. *arXiv preprint arXiv:2210.07774*.

Tatsuru Higurashi, Hayato Kobayashi, Takeshi Masuyama, and Kazuma Murao. 2018. Extractive headline generation based on learning to rank for community question answering. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1742–1753, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

David Hull. 1993. Using statistical testing in the evaluation of retrieval experiments. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '93, page 329–338, New York, NY, USA. Association for Computing Machinery.

Joseph Irwin, Mamoru Komachi, and Yuji Matsumoto. 2011. Narrative schema as world knowledge for coreference resolution. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 86–92, Portland, Oregon, USA. Association for Computational Linguistics.

Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. 2021. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2.

Kalervo Järvelin and Jaana Kekäläinen. 2017. IR evaluation methods for retrieving highly relevant documents. In *ACM SIGIR Forum*, volume 51, pages 243–250. ACM New York, NY, USA.

Serena Jeblee and Graeme Hirst. 2018. Listwise temporal ordering of events in clinical notes. In *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*, pages 177–182.

Heng Ji, Cynthia Rudin, and Ralph Grishman. 2006. Re-ranking algorithms for name tagging. In *Proceedings of the Workshop on Computationally Hard Problems and Joint Inference in Speech and Language Processing*, pages 49–56.

Yunjie Ji, Yan Gong, Yiping Peng, Chao Ni, Peiyan Sun, Dongyu Pan, Baochang Ma, and Xiangang Li. 2023. Exploring ChatGPT's ability to rank content: A preliminary study on consistency with human preferences. *arXiv preprint arXiv:2303.07610*.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023. LLM-Blender: Ensembling large language models with pairwise ranking and generative fusion. *arXiv preprint arXiv:2306.02561*.

Feng Jin, Minlie Huang, and Xiaoyan Zhu. 2010. A comparative study on ranking and selection strategies for multi-document summarization. In *Coling 2010: Posters*, pages 525–533, Beijing, China. Coling 2010 Organizing Committee.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142.

Denys Katerenchuk and Andrew Rosenberg. 2016. RankDCG: Rank-ordering evaluation measure. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3675–3680, Portorož, Slovenia. European Language Resources Association (ELRA).

Sawa Kourogi, Hiroyuki Fujishiro, Akisato Kimura, and Hitoshi Nishikawa. 2015. Identifying attractive news headlines for social media. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1859–1862.

Lakshya Kumar and Sagnik Sarkar. 2022. ListBERT: Learning to rank e-commerce products with listwise BERT. *arXiv preprint arXiv:2206.15198*.

Pawan Kumar, Dhanajit Brahma, Harish Karnick, and Piyush Rai. 2020. Deep attentive ranking networks for learning to order sentences. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8115–8122.

Saar Kuzi, William Cope, Duncan Ferguson, Chase Geigle, and ChengXiang Zhai. 2019. Automatic assessment of complex assignments using topic models. In *Proceedings of the Sixth (2019) ACM Conference on Learning@ Scale*, pages 1–10.

Mirella Lapata. 2006. Automatic evaluation of information ordering: Kendall's tau. *Computational Linguistics*, 32(4):471–484.

Robert Leaman, Rezarta Islamaj Doğan, and Zhiyong Lu. 2013. DNorm: disease name normalization with pairwise learning to rank. *Bioinformatics*, 29(22):2909–2917.

Ann Lee, Michael Auli, and Marc'Aurelio Ranzato. 2021. Discriminative reranking for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7250–7264, Online. Association for Computational Linguistics.

Justin Lee and Sowmya Vajjala. 2022. A neural pairwise ranking model for readability assessment. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3802–3813, Dublin, Ireland. Association for Computational Linguistics.

Hang Li. 2014. Learning to rank for information retrieval and natural language processing. *Synthesis lectures on human language technologies*, 7(3).

Maoxi Li, Aiwen Jiang, and Mingwen Wang. 2013. Listwise approach to learning to rank for automatic evaluation of machine translation. In *Proceedings of Machine Translation Summit XIV: Papers*.

Maoxi Li and Mingwen Wang. 2018. Optimizing automatic evaluation of machine translation with the ListMLE approach. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 18(1):1–18.

Chen Liang, Xiao Yang, Neisarg Dave, Drew Wham, Bart Pursel, and C. Lee Giles. 2018. Distractor generation for multiple choice questions using learning to rank. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 284–290, New Orleans, Louisiana. Association for Computational Linguistics.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Alexander Cosgrove, Christopher D Manning, Christopher Re, Diana Acosta-Navas, Drew Arad Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue WANG, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri S. Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Andrew Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2023. Holistic evaluation of language models. *Transactions on Machine Learning Research*. Featured Certification, Expert Certification.

Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2021. Pretrained transformers for text ranking: BERT and beyond. *Synthesis Lectures on Human Language Technologies*, 14(4):1–325.

Kevin Hsin-Yih Lin and Hsin-Hsi Chen. 2008. Ranking reader emotions using pairwise loss minimization and emotional distribution regression. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 136–144.

Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, Antonios Anastasopoulos, Patrick Littell, and Graham Neubig. 2019. Choosing transfer languages for cross-lingual learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3125–3135, Florence, Italy. Association for Computational Linguistics.

Jiduan Liu, Jiahao Liu, Qifan Wang, Jingang Wang, Wei Wu, Yunsen Xian, Dongyan Zhao, Kai Chen, and Rui Yan. 2023. RankCSE: Unsupervised sentence representations learning via learning to rank. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13785–13802, Toronto, Canada. Association for Computational Linguistics.

Jun Liu, Hiroyuki Shindo, and Yuji Matsumoto. 2018. Sentence suggestion of Japanese functional expressions for Chinese-speaking learners. In *Proceedings of ACL 2018, System Demonstrations*, pages 56–61, Melbourne, Australia. Association for Computational Linguistics.

Tie-Yan Liu et al. 2009. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331.

Xingxian Liu and Yajing Xu. 2023. Learning to rank utterances for query-focused meeting summarization. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8496–8505, Toronto, Canada. Association for Computational Linguistics.

Annie Louis and Mirella Lapata. 2015. Which step do I take first? Troubleshooting with Bayesian models. *Transactions of the Association for Computational Linguistics*, 3:73–85.

Kun Lu, Jin Mao, Gang Li, and Jian Xu. 2016. Recognizing reference spans and classifying their discourse facets. In *Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL)*, pages 139–145.

Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2023a. Fine-tuning LLaMA for multi-stage text retrieval. *arXiv preprint arXiv:2310.08319*.

Xueguang Ma, Xinyu Zhang, Ronak Pradeep, and Jimmy Lin. 2023b. Zero-shot listwise document reranking with a large language model. *arXiv preprint arXiv:2305.02156*.

Yi Ma, Eric Fosler-Lussier, and Robert Lofthus. 2012. Ranking-based readability assessment for early primary children's literature. In *Proceedings of the 2012 Conference of the North American Chapter of the*

Association for Computational Linguistics: Human Language Technologies, pages 548–552.

Ansel MacLaughlin and David A Smith. 2021. Content-based models of quotation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2296–2314.

Nitin Madnani, Rebecca Passonneau, Necip Fazil Ayan, John Conroy, Bonnie Dorr, Judith Klavans, Dianne O'Leary, and Judith Schlesinger. 2007. Measuring variability in sentence ordering for news summarization. In *Proceedings of the Eleventh European Workshop on Natural Language Generation (ENLG 07)*, pages 81–88, Saarbrücken, Germany. DFKI GmbH.

Rana Malhas, Marwan Torki, and Tamer Elsayed. 2016. QU-IR at SemEval 2016 task 3: Learning to rank on Arabic community question answering forums with word embedding. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 866–871, San Diego, California. Association for Computational Linguistics.

Alessandro Mazzei and Valerio Basile. 2019. The DipInfoUniTo realizer at SRST'19: Learning to rank and deep morphology prediction for multilingual surface realization. In *Proceedings of the 2nd Workshop on Multilingual Surface Realisation (MSR 2019)*, pages 81–87, Hong Kong, China. Association for Computational Linguistics.

Paul McNamee, James Mayfield, Dawn Lawrie, Douglas Oard, and David Doermann. 2011. Cross-language entity linking. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 255–263, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Do June Min, Verónica Pérez-Rosas, Kenneth Resnicow, and Rada Mihalcea. 2022. PAIR: Prompt-aware margIn ranking for counselor reflection scoring in motivational interviewing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 148–158, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Michael Mohler, Razvan Bunescu, and Rada Mihalcea. 2011. Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 752–762, Portland, Oregon, USA. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1747–1759, New Orleans, Louisiana. Association for Computational Linguistics.

Jan Niehues, Quoc Khanh Do, Alexandre Allauzen, and Alex Waibel. 2015. ListNET-based MT rescoring. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 248–255.

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

Eric W. Noreen. 1989. *Computer Intensive Methods for Hypothesis Testing: An Introduction*. Wiley.

Gustavo Paetzold and Lucia Specia. 2017. Lexical simplification with neural ranking. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 34–40, Valencia, Spain. Association for Computational Linguistics.

Javier Parapar, David E Losada, and Álvaro Barreiro. 2021. Testing the tests: simulation of rankings to compare statistical significance tests in information retrieval evaluation. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, pages 655–664.

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

Karl Pichotta and John DeNero. 2013. Identifying phrasal verbs using many bilingual corpora. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 636–646, Seattle, Washington, USA. Association for Computational Linguistics.

George Sebastian Pirtoaca, Traian Rebedea, and Stefan Ruseti. 2019. Answering questions by learning to rank - learning to rank by answering questions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2531–2540, Hong Kong, China. Association for Computational Linguistics.

Emily Pitler and Ani Nenkova. 2008. Revisiting readability: A unified framework for predicting text quality. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 186–195, Honolulu, Hawaii. Association for Computational Linguistics.

E. J. G. Pitman. 1937. Significance tests which may be applied to samples from any populations. *Supplement to the Journal of the Royal Statistical Society*.

Vinodkumar Prabhakaran, Ashima Arora, and Owen Rambow. 2014. Staying on topic: An indicator of power in political debates. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1481–1486.

Vinodkumar Prabhakaran, Ajita John, and Dorée D. Seligmann. 2013. Who had the upper hand? Ranking participants of interactions based on their relative power. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 365–373, Nagoya, Japan. Asian Federation of Natural Language Processing.

Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023a. RankVicuna: Zero-shot listwise document reranking with open-source large language models. *arXiv preprint arXiv:2309.15088*.

Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023b. RankZephyr: Effective and robust zero-shot listwise reranking is a breeze! *arXiv preprint arXiv:2312.02724*.

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, and Michael Bendersky. 2023. Large language models are effective text rankers with pairwise ranking prompting. *arXiv preprint arXiv:2306.17563*.

Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. 2008. Learning diverse rankings with multi-armed bandits. In *Proceedings of the 25th international conference on Machine learning*, pages 784–791.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Sebastian Raschka. 2018. Model evaluation, model selection, and algorithm selection in machine learning. *arXiv preprint arXiv:1811.12808*.

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

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

Stefan Riezler and Michael Hagmann. 2021. Validity, reliability, and significance: Empirical methods for NLP and data science. *Synthesis Lectures on Human Language Technologies*, 14(6):1–165.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Erfan Sadeqi Azer, Daniel Khashabi, Ashish Sabharwal, and Dan Roth. 2020. Not all claims are created equal: Choosing the right statistical approach to assess hypotheses. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5715–5725, Online. Association for Computational Linguistics.

Anik Saha, Oktie Hassanzadeh, Alex Gittens, Jian Ni, Kavitha Srinivas, and Bulent Yener. 2023. Improving neural ranking models with traditional ir methods. *arXiv preprint arXiv:2308.15027*.

Tetsuya Sakai. 2014. Statistical reform in information retrieval? *SIGIR Forum*, 48(1):3–12.

Mark Sanderson and Justin Zobel. 2005. Information retrieval system evaluation: Effort, sensitivity, and reliability. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, page 162–169, New York, NY, USA. Association for Computing Machinery.

Cícero dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying relations by ranking with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 626–634, Beijing, China. Association for Computational Linguistics.

Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 373–382.

Libin Shen, Anoop Sarkar, and Franz Josef Och. 2004. Discriminative reranking for machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 177–184, Boston, Massachusetts, USA. Association for Computational Linguistics.

Gabriella Skitalinskaya, Jonas Klaff, and Henning Wachsmuth. 2021. Learning from revisions: Quality assessment of claims in argumentation at scale. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1718–1729, Online. Association for Computational Linguistics.

Niklas Stoehr, Pengxiang Cheng, Jing Wang, Daniel Preotiuc-Pietro, and Rajarshi Bhowmik. 2023. Unsupervised contrast-consistent ranking with language models. *arXiv preprint arXiv:2309.06991*.

Weiwei Sun, Zheng Chen, Xinyu Ma, Lingyong Yan, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023a. Instruction distillation makes large language models efficient zero-shot rankers. *arXiv preprint arXiv:2311.01555*.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Pengjie Ren, Dawei Yin, and Zhaochun Ren. 2023b. Is ChatGPT good at search? Investigating large language models as re-ranking agent. *EMNLP*.

György Szarvas, Róbert Busa-Fekete, and Eyke Hüllermeier. 2013. Learning to rank lexical substitutions. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1926–1932.

Kumiko Tanaka-Ishii, Satoshi Tezuka, and Hiroshi Terada. 2010. Sorting texts by readability. *Computational linguistics*, 36(2):203–227.

Raphael Tang, Xinyu Zhang, Xueguang Ma, Jimmy Lin, and Ferhan Ture. 2023. Found in the middle: Permutation self-consistency improves listwise ranking in large language models. *arXiv preprint arXiv:2310.07712*.

Nadi Tomeh, Nizar Habash, Ryan Roth, Noura Farra, Pradeep Dasigi, and Mona Diab. 2013. Reranking with linguistic and semantic features for Arabic optical character recognition. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 549–555, Sofia, Bulgaria. Association for Computational Linguistics.

Oanh Thi Tran, Bach Xuan Ngo, Minh Le Nguyen, and Akira Shimazu. 2011. A listwise approach to coreference resolution in multiple languages. In *Proceedings of the 25th Pacific Asia Conference on Language, Information and Computation*, pages 400–409, Singapore. Institute of Digital Enhancement of Cognitive Processing, Waseda University.

Kateryna Tymoshenko, Daniele Bonadiman, and Alessandro Moschitti. 2017. Ranking kernels for structures and embeddings: A hybrid preference and classification model. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 897–902.

Shaurya Uppal, Ambikesh Jayal, and Anuja Arora. 2019. Pairwise reviews ranking and classification for medicine e-commerce application. In *2019 Twelfth International Conference on Contemporary Computing (IC3)*, pages 1–6.

Sowmya Vajjala and Detmar Meurers. 2016. Readability-based sentence ranking for evaluating text simplification. *arXiv preprint arXiv:1603.06009*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.

Ellen M Voorhees and Donna K Harman. 2005. *TREC: Experiment and evaluation in information retrieval*. MIT Press.

Feixiang Wang, Zhihua Zhang, and Man Lan. 2016. ECNU at SemEval-2016 task 7: An enhanced supervised learning method for lexicon sentiment intensity ranking. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 491–496, San Diego, California. Association for Computational Linguistics.

Liang Wang, Sujian Li, Yajuan Lv, and Houfeng Wang. 2017. Learning to rank semantic coherence for topic segmentation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1340–1344, Copenhagen, Denmark. Association for Computational Linguistics.

Yifan Wang, Jing Zhao, Junwei Bao, Chaoqun Duan, Youzheng Wu, and Xiaodong He. 2022. LUNA: Learning slot-turn alignment for dialogue state tracking. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3319–3328, Seattle, United States. Association for Computational Linguistics.

Frank Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometrics Bulletin*.

Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, Hui Xiong, and Enhong Chen. 2023. A survey on large language models for recommendation. *arXiv preprint arXiv:2305.19860*.

Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. 2008. Listwise approach to learning to rank: theory and algorithm. In *Proceedings of the 25th international conference on Machine learning*, pages 1192–1199.

Yan Yan, Bo-Wen Zhang, Xu-Feng Li, and Zhenhan Liu. 2020. List-wise learning to rank biomedical question-answer pairs with deep ranking recursive autoencoders. *PloS one*, 15(11):e0242061.

Ruosong Yang, Jiannong Cao, Zhiyuan Wen, Youzheng Wu, and Xiaodong He. 2020. Enhancing automated essay scoring performance via fine-tuning pre-trained language models with combination of regression and ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1560–1569, Online. Association for Computational Linguistics.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading ESOL texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189, Portland, Oregon, USA. Association for Computational Linguistics.

Wenpeng Yin, Lifu Huang, Yulong Pei, and Lian'en Huang. 2012. RelationListwise for query-focused multi-document summarization. In *Proceedings of COLING 2012*, pages 2961–2976, Mumbai, India. The COLING 2012 Organizing Committee.

Sina Zarrieß, Aoife Cahill, and Jonas Kuhn. 2012. To what extent does sentence-internal realisation reflect discourse context? A study on word order. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 767–776, Avignon, France. Association for Computational Linguistics.

Sina Zarrieß and Jonas Kuhn. 2013. Combining referring expression generation and surface realization: A corpus-based investigation of architectures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1547–1557, Sofia, Bulgaria. Association for Computational Linguistics.

Yue Zhang, ChengCheng Hu, Yuqi Liu, Hui Fang, and Jimmy Lin. 2021. Learning to rank in the age of Muppets: Effectiveness–efficiency tradeoffs in multi-stage ranking. In *Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing*, pages 64–73, Virtual. Association for Computational Linguistics.

Zongmeng Zhang, Wengang Zhou, Jiaxin Shi, and Houqiang Li. 2023. Hybrid and collaborative passage reranking. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 14003–14021, Toronto, Canada. Association for Computational Linguistics.

Zhicheng Zheng, Fangtao Li, Minlie Huang, and Xiaoyan Zhu. 2010. Learning to link entities with knowledge base. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 483–491, Los Angeles, California. Association for Computational Linguistics.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zhicheng Dou, and Ji-Rong Wen. 2024. Large language models for information retrieval: A survey. *arXiv preprint arXiv:2308.07107*.

Honglei Zhuang, Zhen Qin, Rolf Jagerman, Kai Hui, Ji Ma, Jing Lu, Jianmo Ni, Xuanhui Wang, and Michael Bendersky. 2022. RankT5: Fine-tuning T5 for text ranking with ranking losses. *arXiv preprint arXiv:2210.10634*.

Shengyao Zhuang, Bing Liu, Bevan Koopman, and Guido Zuccon. 2023. Open-source large language models are strong zero-shot query likelihood models for document ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8807–8817, Singapore. Association for Computational Linguistics.

Appendices

A Selecting papers

We searched the ACL Anthology for query terms involving popular LTR algorithms such as SVMrank, ListNet, ListMLE9 and AdaRank, and using the queries "learning to rank" and "learning-to-rank". Among the results, we excluded papers that discussed the classic information retrieval task (search, crosslingual information retrieval, etc.), and selected papers with the goal of representing diverse NLP tasks where LTR methods are used. Vision-language tasks are also not included. Other non-NLP venues (e.g., CIKM, SIGIR, PlosOne, etc.) also sometimes report on research that employes LTR methods on NLP tasks, and we included them where relevant, based on Google Scholar result for the same queries.

B Tabulated surveyed papers

In performing this review we tabulated a range of information about the $\sim$ 150 papers surveyed. Some statistics throughout the paper are generated using information in this spreadsheet. It is too large to fit in a paper format, but we make it available here: `https://github.com/nishkalavallabhi/LTRSurvey2024`.

C Details on usage of significance testing

The most frequently used test was the paired t-test, which was applied in 15 papers to a wide variety of metrics including precision, recall, F-score, MAP, generalized average precision, P@K, MRR@k, NDCG, correlation measures, perplexity and metrics used for coreference resolution (Xia et al., 2008; Irwin et al., 2011; Yannakoudakis et al., 2011; Gupta and Kumaraguru, 2012; Szarvas et al., 2013; Severyn and Moschitti, 2015; Nogueira et al., 2020; Yan et al., 2020; Amiri et al., 2021; Azarbonyad et al., 2021; Skitalinskaya et al., 2021; Zhang et al., 2021; Frydenlund et al., 2022; Tang et al., 2023; Zhuang et al., 2023). The second-most frequent was Wilcoxon's test, which was similarly applied to many different metrics, in a total of six papers (Burges et al., 2005; Jin et al., 2010; Chen and Ji, 2011; Louis and Lapata, 2015; Higurashi et al., 2018; Lee and Vajjala, 2022). Additionally, McNamee et al. (2011) applied the sign test to P@1, Liang et al. (2023) used the paired bootstrap on various metrics, Burges et al. (2006) reported the overlap of confidence intervals on NDCG, and Narayan et al. (2018) conducted one-way ANOVA with post-hoc Tukey HSD tests on the distribution of ranks. An un(der)specified test was used in a further 11 papers (Lin and Chen, 2008; Tran et al., 2011; Ma et al., 2012; Zarrieß and Kuhn, 2013; Vajjala and Meurers, 2016; Li and Wang, 2018; Min et al., 2022; Rau and Kamps, 2022; Zhuang et al., 2022; Liu et al., 2023; Liu and Xu, 2023).