A Linguistic Study on Relevance Modeling in Information Retrieval

Yixing Fan, Jiafeng Guo, Xinyu Ma, Ruqing Zhang, Yanyan Lan, and Xueqi Cheng

CAS Key Lab of Network Data Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China

ABSTRACT
Relevance plays a central role in information retrieval (IR), which has received extensive studies starting from the 20th century. The definition and the modeling of relevance has always been critical challenges in both information science and computer science research areas. Along with the debate and exploration on relevance, IR has already become a core task in many real-world applications, such as Web search engines, question answering systems, conversational bots, and so on. While relevance acts as a unified concept in all these retrieval tasks, the inherent definitions are quite different due to the heterogeneity of these tasks. This raises a question to us: Do these different forms of relevance really lead to different modeling focuses? To answer this question, in this work, we conduct an empirical study on relevance modeling in three representative IR tasks, i.e., document retrieval, answer retrieval, and response retrieval. Specifically, we attempt to study the following two questions: 1) Does relevance modeling in these tasks really show differences in terms of natural language understanding (NLU)? We employ 16 linguistic tasks to probe a unified retrieval model over these three retrieval tasks to answer this question. 2) If there do exist differences, how can we leverage the findings to enhance the relevance modeling? We proposed three intervention methods to investigate how to leverage different modeling focuses of relevance to improve these IR tasks. We believe the way we study the problem as well as our findings would be beneficial to the IR community.

CCS CONCEPTS
• Computer systems organization → Embedded systems; Redundancy; Robotics; • Networks → Network reliability.

KEYWORDS
relevance modeling, information retrieval

ACM Reference Format:
Yixing Fan, Jiafeng Guo, Xinyu Ma, Ruqing Zhang, Yanyan Lan, and Xueqi Cheng. 2021. A Linguistic Study on Relevance Modeling in Information Retrieval. In Proceedings of the Web Conference 2021 (WWW ‘21), April 19–23, 2021, Ljubljana, Slovenia. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3442381.3450009

1 INTRODUCTION
Information retrieval (IR) has already became a ubiquitous activity in our daily life. People rely on IR systems to obtain information that is relevant to their needs. Relevance, which denotes how well a retrieved document meets the information need of a user, plays a central role in IR. In fact, all the retrieval models in IR systems are trying to approximate the relevance from the perspective of users. However, the concept of relevance, likes all the other human notions, is an open and vague subject [45].

It has been a long-standing challenge to understand and model relevance in two major research communities, i.e., information science community and computer science community. On one hand, researchers from information science community studied the definition of relevance concept since 1950s [18, 28, 44]. They tried to uncover the aspects of the relevance based on the data collected from tests or questionnaires. On the other hand, researchers from computer science community mainly focused on the modeling/computation of relevance since the mid-1960s [26]. A large number of models have been proposed to evaluate the relevance degree of a document with respect to users’ information needs [14, 23]. These models have evolved from shallow to deep understanding of the document and the information need, which are often based on heuristically designed features or functions. However, there has been few studies to take the relevance definition into account in designing relevance models.

Along with the debate and exploration on relevance, IR has been widely applied and become a core task in many real-world applications, such as Web search engines, question answering systems, conversational bots, and so on. In Web search engines, the IR task is to rank a list of documents according to their relevance to a given user query. In question answering systems, the IR task is to retrieve a few relevant answers from the archived answer pool with respect to a user’s question. In conversational bots, the IR task is to find the relevant response from existing human-generated conversation repository as the reply to the input utterance. Without loss of generality, relevance acts as a unified concept in all these IR tasks. However, we may find subtle differences on the definition of the relevance concept among these tasks. For example, the relevant documents in Web search often means topical relevance to the search query [23]. The relevant answers in question answering need to correctly address the question [30]. Finally, the relevant responses in conversation actually refer to some kind of correspondence with respect to the input utterance [25]. In summary, the inherent definitions of relevance actually are quite different due to the heterogeneity of different IR tasks [13].
The above observations naturally raise a question to us: Do different forms of relevance in these IR tasks really lead to different modeling focuses? To answer this question, in this paper, we conduct an empirical study to investigate the relevance modeling in three representative IR tasks, namely document retrieval, answer retrieval, and response retrieval. More specifically, we break down the study into the following two concrete research questions:

- **RQ1**: Since these tasks are all text based, does relevance modeling in different IR tasks really show differences in terms of natural language understanding?
- **RQ2**: If there do exist differences, how can we leverage these findings to enhance the relevance modeling on each IR task?

For the first question, we propose to leverage the probe-based method, which has been widely adopted in understanding the language modeling [7, 24], to analyze the potential differences in relevance modeling in the three IR tasks. Towards this goal, there are two basic requirements for the design of our empirical experiments: 1) It is better to have a unified IR model which can perform well on all these IR tasks, so that we can form a fair comparison basis. 2) The model should be able to integrate a variety of probing tasks, so that we can compare the modeling focuses easily. To meet these requirements, we take the recently proposed Bert model [10], which has obtained reasonably good performances on these three retrieval tasks [6, 8, 31], as the unified IR model for study. We then utilize 16 probing tasks related to language modeling to compare the differences of relevance modeling in the three IR tasks from the language understanding perspective. For the second question, we utilize the intervention method to study how to enhance the relevance modeling in different IR tasks based on the previous findings. The basic idea is to interfere an existing relevance model with each probe task as an intervention factor to see how the performance varied on each retrieval task.

Through the above experiments, our analysis reveals the following interesting results:

- For RQ1: The answer is YES. The three IR tasks show different modeling focuses on relevance from the natural language understanding view. Specifically, the document retrieval focuses more on semantic tasks, the answer retrieval pays attention to both syntactic and semantic tasks, while the response retrieval has little preference to most of the linguistic tasks. Beyond these differences, The understanding of the Synonym seems universally useful for all the three retrieval tasks.
- Furthermore, we also find that there are different language understanding requirements for the two inputs in relevance models. A by-product is that we can thus analyze the *inherent heterogeneity* of the IR task by comparing its modeling focuses on the two inputs. Through our analysis, it is interesting to find that the answer retrieval is the most heterogeneous one rather than the document retrieval which is often considered heterogeneous based on its surface form [13].
- For RQ2: We demonstrate that we are able to improve the relevance modeling based on the above findings by the parameter intervention method.

The rest of the paper is organized as follows: In Section 2, we describe the representative retrieval tasks in IR. We then present the probing analysis and intervention analysis in Section 3 and Section 4, respectively. The Section 5 discuss the related work while conclusions are made in Section 6.

## 2 RETRIEVAL TASKS IN INFORMATION RETRIEVAL

In this section, we introduce the IR tasks used in this work for the relevance modeling analysis. Given a user’s information need \( S \) (e.g., query, utterance, or question), a retrieval task aims to find relevant information \( T = \{t_1, t_2, ..., t_k\} \) (e.g., Web pages, response, and answers) from an archived information resources \( D \). The retrieved documents are returned as a ranking list through a ranking model according to their relevance degree to the input query. A major characteristic of document retrieval is the length heterogeneity between queries and documents. The user queries are often very short with unclear intents, consisting of only several key words in most cases. Existing works have shown that the average length of queries is about 2.35 tokens [48]. However, the documents are usually collected from the World Wide Web and have longer text lengths, ranging from multiple sentences to several paragraphs. This heterogeneity leads to the typical vocabulary mismatching problem, which has long been a challenge in the relevance modeling of document retrieval [23]. To address this issue, a great amount of efforts has been devoted to design effective retrieval models to capture the semantic matching signals between the query and the document for document retrieval [13, 23].

### 2.1 Document Retrieval

Document retrieval is a classical task in IR [50], which has been widely used in modern Web search engines, such as Google, Bing, Yandex, and so on. In this task, users typically specify their information needs via a query \( Q \) to an information system to obtain the relevant documents \( D \). The retrieved documents are returned as a ranking list through a ranking model according to their relevance degree to the input query. A major characteristic of document retrieval is the length heterogeneity between queries and documents. The user queries are often very short with unclear intents, consisting of only several key words in most cases. Existing works have shown that the average length of queries is about 2.35 tokens [48]. However, the documents are usually collected from the World Wide Web and have longer text lengths, ranging from multiple sentences to several paragraphs. This heterogeneity leads to the typical vocabulary mismatching problem, which has long been a challenge in the relevance modeling of document retrieval [23]. To address this issue, a great amount of efforts has been devoted to design effective retrieval models to capture the semantic matching signals between the query and the document for document retrieval [13, 23].

### 2.2 Answer Retrieval

Answer retrieval is widely used in question answering (QA) systems, such as StackOverflow 1, Quora 2, and Baidu Zhidao 3. The QA system directly retrieves the answer \( A \) to the question \( Q \) from existing answer repository \( T \). The core of the QA system is to compute relevance scores between questions and candidate answers, and subsequently ranking them according to the score. Compared with document retrieval, answer retrieval is more homogeneous and poses different challenges. Specifically, the questions are usually natural language, which are well-formed sentence(s) and have clearer intent description. While the answers are usually shorter text spans, e.g., sentences or passages, which have more concentrated topics. However, answer retrieval is still a challenge problem since an answer should not only be topically related to but also correctly address the question. Different retrieval models have been

---

1. https://stackoverflow.com/
2. https://quora.com/
3. https://zhidao.baidu.com/
propose for the answer retrieval. Earlier statistical approaches focused on complex feature engineering, e.g., lexical and syntactic features [58]. In recent years, end-to-end neural models have been applied for relevance modeling in answer retrieval and achieved state-of-the-art performances [13].

2.3 Response Retrieval

Response retrieval is a core task in automatic conversation systems, such as Apple Siri, Google Now, and Microsoft XiaoIce. The conversation system relies on response retrieval to select a proper response $R$ from a dialog repository $T$ with respect to an input utterance $U$. In multi-turn response retrieval, there is a context $C$ accomplished with each utterance $U$, where the context contains the conversation histories before the utterance. Different from document retrieval and answer retrieval, the input utterance and candidate responses are often short sentence, which are homogeneous in the form. The relevance in response retrieval often refers to certain semantic correspondence (or coherent structure) which is broad in definition, e.g., given an input utterance "OMG I got myopia at such an 'old' age", the response could range from general (e.g., "Really?") to specific (e.g., "Yeah. Wish a pair of glasses as a gift") [54]. Therefore, it is often critical to model the coherence and avoid general trivial responses in response retrieval. In recently years, researchers have proposed a variety of approaches for response retrieval tasks [5], where the neural network based methods have achieved state-of-the-art performance [6].

3 PROBING ANALYSIS

In this section, we aim to address the first research question, that is, whether the relevance modeling in different IR tasks really shows differences in terms of natural language understanding. For this purpose, we propose to leverage the probing-based method to analyze the potential differences in the relevance modeling of the above three IR tasks. In the following, we will give the detailed description of the analysis process, including the probing method, the probing tasks, and the experimental results.

3.1 The Probing Method

The core idea of the probing analysis is to learn a unified representative retrieval model over the three IR tasks, and probe the learned model to compare the focuses between different relevance modeling tasks. Specifically, we take the recently proposed Bert model as the unified retrieval model since it has obtained reasonably good performances on all the retrieval tasks [8, 31, 52]. Moreover, the Bert model is a stack of multiple Transformer layers [10], which can easily integrate different probing tasks on each Transformer layer. In this way, we could investigate the nuanced requirements of relevance modeling, and form a fair comparison between different IR tasks.

To learn the retrieval model for each IR task, we finetune the original Bert model to achieve good performances on each retrieval dataset respectively. We then probe the original Bert and the finetuned Bert with a set of natural language understanding tasks [3, 24]. Specifically, for the model to be probed, either the original or the finetuned Bert, we take an additional multi-layer perceptron (MLP) as the prediction layer over the target layers to be probed. We then train and evaluate the probing task over the model to assess its ability in capturing the corresponding linguistic properties. It is worth to note that the Bert layers are fixed during the probing, since we aim to investigate what have been encoded in these layers. Finally, we analyze the performance gap of each probing task between the original and finetuned Bert over each IR task. Note that it is improper to directly compare the absolute performance of the finetuned Bert models on different IR tasks since the training corpus varies a lot. On the contrary, by taking the original Bert as a baseline, the relative performance gap of the finetuned Bert over the baseline on a probing task could reflect the importance of the specific linguistic property for the corresponding retrieval task.

3.2 Probing Tasks

We utilize a suite of 16 diverse probing tasks related to natural language understanding to investigate the focuses of the retrieval model, including lexical tasks, syntactic tasks, and semantic tasks. Here, most of the probing tasks have been utilized to study the linguistic properties of neural language models in different NLP tasks [3, 24], e.g., language model [7], sentence embedding [34], natural language inference [39]. In this work, we take them to study the preferences of the relevance modeling in each retrieval task. In addition, we also introduce four probing tasks, which are closely related to the semantic matching between natural sentences, i.e., synonym identification, polysemy identification, keyword extraction, and topic classification. In the following, we will describe each probing task in detail, and the statistics of the datasets and the settings are listed in the Appendix C.

3.2.1 Lexical Tasks. The lexical tasks focus on the lexical meaning and positions of a term in sentences, paragraphs, or documents. It lies at the low level of the natural language understanding [37]. Here, we take three typical lexical tasks for the probing.

The Text Chunking (Chunk) task, also referred to as shallow parsing, aims to divide a complicated text into smaller parts. This task assesses whether the relevance modeling captures the notions of the spans and boundaries. We use the CoNLL 2000 dataset [42] for experiments.

The Part-of-Speech Tagging (POS) task is the process of marking up of words in a sentence as nouns, verbs, adjectives, adverbs etc. This task tests whether the relevance modeling captures the POS knowledge. Here, we take UD-EWT dataset [47] for experiments.

The Named Entity Recognition (NER) is the task of identifying entities and their category from a given text. This task assesses whether the relevance modeling pay attention to the entity information. We use the CoNLL 2003 dataset [43] for experiments.

3.2.2 Syntactic Tasks. The syntactic task is on the linguistic discipline dealing with the relationships between words in a sentence (i.e. clauses), the correct creation of the sentence structure and the word order.

The Grammatical Error Detection (GED) task is to detect grammatical errors in sentence. It is to assess whether the grammatical information is required for the relevance modeling. We use the First Certificate in English dataset [46] for experiments.

The Syntactic Dependence task is to examine whether the syntactic relationships between words are crucial to model the
relevance. We follow the work [24] to take arc prediction and
classification for experiments. Specifically, the syntactic arc
dependency prediction (SynArcPred) is a binary classification task,
which aims to identify whether a relation exists between two tokens.
The syntactic arc dependency classification (SynArcCls) is a multi-
class classification task, which assumes the input tokens is linked
with each other and identifies which relationship it is. We use the
UD-EWT datasets [47] for experiments.

The Word Scramble is a binary classification task which as-

seSSes whether the word order and structure affects the mean-
ing of a sentence. We use it to test whether the relevance modeling
cares about the word orders of the sentences/documents. We use the
PAWS-wiki dataset [62] for experiments.

3.2.3 Semantic Tasks. The semantic tasks deal with the semantic
meaning of the words and sentences, the ways that words and
sentences refer to each other. It lies at the high level of text un-
derstanding.

The Preposition Supersense Disambiguation is to examine
whether semantic contribution of preposition is important factors to
model the relevance. We follow previous work [24] to take two sub-
tasks for experiments, namely PS-fxn and PS-role. Specifically, the
PS-fxn concerns the function of the preposition, while the PS-role
determines the role of the preposition. We use the STREUSLE 4.0
corpus [46] for experiments.

The coreference arc prediction (CorefArcPred) is to assess
whether two mentions share the same coreference cluster. We use it
to test whether the relevance modeling captures the coreference
relationship between pronouns and entities. We use the CoNLL
dataset [36] for experiments.

The Semantic Dependence task is to assess whether the se-
matic relationships between words are important for the rele-
ance modeling. We follow the work [24] to take arc prediction
and arc classification for experiments. Specifically, the semantic arc
dependency prediction (SemArcPred) aims to identify whether a
semantic relation exits between two tokens. The semantic arc de-
pendency classification (SemArcCls) assumes the input tokens is
linked with each other and identifies which semantic relationship
it is. We use the SemEval 2015 dataset [32] for experiments.

The Synonym and the Polysemy task deal with the semantic
meaning of a word pair from two sentences. The synonym focus on
identifying whether two different words from similar context share
the same meaning, while the polysemy is to distinguish the meaning
of the same word from two sentences. We use them to test whether
the relevance modeling captures the semantic meaning between
word pairs. For these two tasks, we crawled 10k sentences from an
online Website for experiments. We will release these datasets after
the paper is accepted.

The Keyword extraction task is to identify the prominent words
that best describe the subject of a document. This task is to test
whether the relevance modeling focuses on the keywords to in-
teract the input pairs. Here, we take the Inspec [20] dataset for
experiments.

The Topic Classification is to classify a document into a pre-
defined topic. We use it to test whether the relevance modeling pay
attention to the topics of the text inputs. Here, we use the Yahoo!

| Tasks         | #S | #T | AvgLen(S) | AvgLen(T) | #Vocab |
|---------------|----|----|-----------|-----------|--------|
| Robust04      | 250| 0.5M| 2.6       | 465       | 27194  |
| MsMarco       | 100K| 1M  | 6.4       | 56.3      | 27636  |
| Ubuntu        | 0.59M| 0.66M| 10.3      | 22.2      | 24026  |

Table 1: Dataset statistics of each retrieval tasks, S represents
the left input of each retrieval task. T represents the right
input of each retrieval task.

For experiments, we first introduce the settings for the retrieval
tasks, including models, datasets, and configurations. Then, we
describe the settings of the probing tasks.

3.3.1 Retrieval model. Here, we take the off-the-shell BERTbase
(Devlin et al., 2018) model, which has been proved to be effective in
many retrieval tasks [8, 31, 52], as the retrieval model in all the three
retrieval tasks. Specifically, the model takes the concatenation of a
text pairs as input with a special token “[SEP]” separating the two
segments. To further separate the left input from the right input,
we follow the work [10] to add two additional tokens “[S]” and “[T]”
into the two segments. All tokens are mapped into an embedding,
with an additional position embedding concatenated in it. Then, the
tokens go through several transformer layers to fully interact with
each other. Finally, the output embedding of the first token is used
as the interaction of the input text pairs, and fed into a multi-layer
perceptron (MLP) to obtain the final relevance score. For fair com-
parison, we directly take the Bert-base model4 (uncased, 12-layer,
768-hidden, 12-heads, 110M parameters) as the implementation as
the retrieval model. For model learning, we leverage the pre-trained
language model released in the original Bert as the initialization and
finetune it on the corresponding datasets. The MLP layer is
learned from scratch as the previous works [10]. For more detailed
settings of all the retrieval models, please refer to Appendix A.

3.3.2 Retrieval Datasets. To learn the retrieval model, we take three representative benchmark dataset, i.e., Robust04 [50], MsMarco [30], and Ubuntu [25], for the relevance modeling in document retrieval, answer retrieval, and response retrieval, respectively. The statistics of these datasets are shown in Table 1. As we can see, these
datasets show very different patterns in terms of the average length
of the text pairs in different tasks. Document retrieval is the most
heterogeneous as the average length of query and document is
2.6 and 465, respectively. While answer retrieval has reduced hetero-
genity compared with document retrieval. The response retrieval is
relatively homogeneous as the average length of the utterance
and response are very close to each other. For all these datasets, we
simply padded each short text pairs with [PAD] and truncated long
text pairs into 512 tokens.

For task evaluation, we take the NDCG@20 for document re-
trieval, MRR@10 in answer retrieval, and recall@1 in response
retrieval, as is done in previous works [8, 31, 52].

4https://github.com/google-research/bert
3.4 Results

In this section, we show the probe experiments as well as the results by answering the following research questions.

3.4.1 How does the unified retrieval model perform on each retrieval task? We take the same original Bert model as the starting point, and finetune it on each IR task to learn task specific requirements for relevance modeling. In the following sections, we will use BERT$_{base}$ to denote the original Bert model, and BERT$_{doc}$, BERT$_{ans}$, and BERT$_{rsp}$ to denote the finetuned Bert on document retrieval, answer retrieval, and response retrieval, respectively. Here, we show the performances of the BERT$_{base}$ as well as the finetuned models on the IR datasets with respect to each layer in Figure 1. The results are summarized as follows.

Firstly, we can see that the BERT$_{base}$ which learned over a large amount of unstructured texts in an unsupervised way, has already achieved good performances on all the three retrieval tasks (The existing state-of-the-art performances on each datasets are listed in appendix B). It indicates that the linguistic information encoded in Bert [24] is useful for relevance modeling. In addition, it is worth to note that the best performance of the BERT$_{base}$ is not always achieved at the last layer, e.g., the answer retrieval on MsMarco gets the best result on the sixth layer. The results suggest that the probing should better be conducted over all the layers, not just the last layer, to select the best-performing layer to study.

Secondly, we can see that the finetuned Bert can significantly improve the performances on all the retrieval tasks. Specifically, the relative improvement of the finetuned Bert (i.e., BERT$_{doc}$, BERT$_{ans}$, and BERT$_{rsp}$) against the BERT$_{base}$ over the best layer is about 23.3%, 45.3%, and 51.6%, respectively. These improvements indicate that the finetuned Bert models are able to learn task specific properties for the relevance modeling on each IR task.

Finally, we can observe that the finetuned Bert achieved larger improvements on the higher layers than the lower layers on all the three tasks, and the last layer always performs the best. This is consistent with the findings of existing work [24] that the higher layers of the finetuned Bert tend to learn the task specific features, while the lower layers learn the basic linguistic features.

3.4.2 Do different IR tasks show different modeling focuses in terms of natural language understanding? Here, we study the differences between IR tasks through quantitative analysis based on the performance gap of the probing tasks. As found in previous section, the best performance of a probing task could be achieved by any layer of the Bert model. For fair comparison, we take the best layer from the BERT$_{base}$ model and the finetuned Bert models (i.e., BERT$_{doc}$, BERT$_{ans}$, and BERT$_{rsp}$) for the following study. The results are summarized in the Table 2.

We first look at each IR task and find the following performance patterns.

1) For document retrieval, there is a clear pattern that the relevance modeling focuses more on the semantic tasks than the lexical and syntactic tasks. The performance gap on most semantic tasks between BERT$_{doc}$ and BERT$_{base}$ is positive and significant. Among them, The top-2 improved tasks are Synonym and Polysemy, showing that relevance modeling in document retrieval requires better understanding of the semantic meaning of a word pair. This is somehow consistent with the previous findings [?] that topic models (e.g. PLSI [?] and LDA [?]), which capture the synonym and polysemy well, can be applied to improve the document retrieval models.

2) For answer retrieval, most probing tasks (i.e., 11 out of 16) have been improved by BERT$_{ans}$, among which eight improvements are significant. It indicates that the relevance modeling in answer retrieval is more difficult, which requires more comprehensive language understanding as compared with the other two. Specifically, BERT$_{ans}$ improves all the syntactic-levels tasks, showing that the syntactic features, like word order and structure in a sentence, are important for relevance modeling in answer retrieval.

3) For response retrieval, it is surprising to see that the performances of most probing tasks (i.e., 12 out of 16) have been decreased by BERT$_{rsp}$, among which ten drops are significant. It suggests that most linguistic properties encoded by the original Bert has already been sufficient for the relevance modeling in response retrieval. Meanwhile, we can find that BERT$_{rsp}$ improves Synonym while decreases Polysemy significantly, as two extremes. The results demonstrate that response retrieval need to better understand similar words.
Table 2: Overall performances of each probing tasks on different retrieval tasks. Significant improvement or degradation with respect to Bert$_{\text{base}}$ is indicated (+/-) (p-value ≤ 0.05 with Bonferroni correction).

| Probing Tasks   | Document Retrieval | Answer Retrieval | Response Retrieval |
|-----------------|---------------------|------------------|--------------------|
|                 | base | Bert$_{\text{doc}}$ | Δ     | base | Bert$_{\text{ans}}$ | Δ     | base | Bert$_{\text{rsp}}$ | Δ     |
| Lexical Tasks   |      |                   |       |      |                   |       |      |                   |       |
| Chunk           | 92.6 | 92.47              | -0.14%| 92.9 | 92.53              | -0.40%| 92.47 | 92.49              | +0.02%|
| POS             | 95.89| 95.72              | -0.18%| 95.45| 95.48              | +0.03%| 95.7  | 95.55              | -0.16%|
| NER             | 83.51| 83.16              | -0.42%| 80.1 | 80.95              | +1.06%*| 82.16 | 80.72              | -1.52%*|
| Syntactic Tasks |      |                   |       |      |                   |       |      |                   |       |
| GED             | 41.83| 40.56              | -3.04%| 41.44| 41.8               | +0.87%| 41.23 | 39.72              | -3.66%*|
| SynArcPred      | 87.29| 87.21              | -0.09%| 86.44| 86.75              | +0.36%| 86.44 | 85.95              | -0.57%*|
| SynArcCls       | 93.66| 93.59              | -0.07%| 93.37| 93.43              | +0.06%| 93.32 | 92.87              | -0.25%|
| Word Scramble   | 62.04| 61.87              | -0.27%| 62.17| 62.87              | +1.13%*| 59.91 | 60.19              | +0.47%*|
| Semantic Tasks  |      |                   |       |      |                   |       |      |                   |       |
| PS-fxn          | 89.21| 89.89              | +0.76% | 87.92| 88.95              | +1.17%*| 89.95 | 86.89              | -3.4%  |
| PS-role         | 78.2 | 79.55              | +1.73% | 77.63| 79.18              | +2.00%*| 79.22 | 80.14              | +1.16%*|
| CorefArcPred    | 78.22| 78.46              | +0.31% | 77.5 | 76.93              | -0.74% | 79.53 | 78.3               | -1.6%  |
| SemArcPred      | 87.34| 86.96              | -0.44% | 87.69| 88.01              | +0.06% | 87.23 | 87.09              | -0.16% |
| SemArcCls       | 92.47| 92.45              | -0.02% | 92.67| 92.43              | -0.43% | 92.98 | 92.33              | -0.7%  |
| Polysemy        | 64.1 | 67.1               | +4.68% | 64.1 | 69.1               | +7.8%  | 64.1  | 58.9               | -11.17%|
| Synonym         | 66.32| 78.49              | +18.35%| 66.33| 75.86              | +14.37%*| 66.31 | 80.68              | +21.67%*|
| Keyword         | 48.66| 48.98              | +0.66% | 48.84| 48.72              | -0.25% | 46.66 | 45.95              | -1.52%*|
| Topic           | 66.93| 67.82              | +1.33% | 67.7 | 69.16              | +2.16%*| 67.34 | 66.11              | -1.83%*|

In different contexts than to distinguish the same words in different context.

We then look at each probing task and obtain the following observations across different IR tasks.

1) The CorefArcPred and Keyword tasks have only been significantly improved by BERT$_{\text{doc}}$ among the three finetuned models but decreased by the rest. Meanwhile, the NER and GED tasks have only been significantly improved by BERT$_{\text{ans}}$ but drop on the other two. The results indicate that relevance modeling in document retrieval pays more attention to similar keywords while the relevance modeling in answer retrieval pays more attention to identifying targeted entities in questions and answers.

2) The Word Scramble task obtains significant improvement by both BERT$_{\text{ans}}$ and BERT$_{\text{rsp}}$, but drops by BERT$_{\text{doc}}$. It suggests that the relevance modeling in both answer retrieval and response retrieval cares more about the word order and sentence structure than that in document retrieval. This also explains why keyword-based methods could work very well for ad-hoc retrieval (i.e., document retrieval). Moreover, the Polysemy and Topic tasks obtain significant improvement by BERT$_{\text{doc}}$ and BERT$_{\text{ans}}$, but drop significantly BERT$_{\text{rsp}}$. In fact, the Polysemy is also connected with topic identification since it aims to identify polysemic words under different topics. This indicates that the relevance modeling in both document retrieval and answer retrieval pays more attention to topic understanding than that in response retrieval.

3) Despite the above differences, there are some common patterns across the three tasks. We can see that both the Synonym and PS-role tasks have been improved significantly by all the three finetuned Bert models. Moreover, the improvement of the Synonym task is always the largest on all the three retrieval tasks. These results demonstrate that it is of great importance to capture the synonyms in all the relevance modeling tasks.

Based on all the above observations, we can conclude that the relevance modeling in the three representative retrieval tasks shows quite different modeling focuses in terms of natural language understanding.

3.4.3 Do relevance modeling treat their inputs differently in terms of natural language understanding? Since relevance models typically take a pair of texts as inputs, we further study the performance gap of each probing task on the left and right input, respectively. Here we directly mask the tokens in the right input when testing the left input, and vice versa. In this study, we only keep the probing tasks whose input is a single sentence, and ignore the tasks that require a pair of inputs (i.e., The Word Scramble, CorefArcPred, Polysemy, and Synonym). Similar as the previous section, we take the best layer in BERT as the representative performance for each comparison. The results are depicted in Figure 2, with the blue and orange bar represent the performance gap between the BERT$_{\text{base}}$ model and the finetuned Bert models on the left and right input, respectively. In the following, we use the term “similar trend” to denote the case where the performance gap are both positive or both negative on the left and right inputs, and use the term “reverse trend” if the gap directions are reverse to each other on the two inputs. Specifically, we have the following observations:

1) In document retrieval, the left (query) side and the right (document) side show similar trends on both lexical and syntactic probing tasks, although the gap sizes are different in most cases. Meanwhile, they also show different modeling focuses on most semantic tasks. Specifically, the query side cares about the coarse-level functions of the prepositions (i.e., PS-fxn) while the document side pays attention to both the coarse-level functions and the fine-grained roles (i.e., PS-fxn and PS-role) in terms of the prepositions. The query
Figure 2: Probing task performance comparison between the left and right input on three retrieval tasks. Each bar denotes the improvement/decrease in the performance over the corresponding baseline. The query/question/utterance and document/answer/response are the left and right input of document retrieval, answer retrieval, and response retrieval, respectively.

2) In answer retrieval, we can see that the left (question) side and the right (answer) side show very different preferences on most probing tasks. The question side improves 5 out of 12 probing tasks, while the answer side improves 7 out of 12 probing tasks. More importantly, they show the reverse trend on half of the probing tasks (i.e., POS, SynArcPred, SynArcCls, PS-role, PS-fxn and SemArcPred). Moreover, we can find that the question side pays more attention to semantic tasks, while the answer side cares more about the lexical and syntactic tasks. The results also indicate that understanding prepositions properly (i.e., PS-role and the PS-fxn) could be of great importance in understanding the question.

3) In response retrieval, the left (utterance) side and the right (response) side show similar trends on most of the probing tasks (i.e. 10 out of 12), with Chunk and PS-role as the exceptions. Among those similar trends, the gap sizes are quite different on the two sides. For example, the utterance side drops more on POS, SynArcPred, Keyword and SemArcPred, while the response side drops more on Chunk, GED, and SynArcCls.

Based on the above results, we can further analyze the inherent heterogeneity of the three retrieval tasks by comparing the linguistic focuses between their left and right inputs. Here we take the reverse trend on the two inputs as a key signal for the inherent heterogeneity, which indicates significantly different modeling focuses on a probing task. As a result, we can find that the answer retrieval (i.e., 6 reverse trends) is the most heterogeneous inherently from the linguistic view, followed by the document retrieval (i.e., 3 reverse trends) and the response retrieval (i.e., 2 reverse trends). This is an interesting result since the previous works [137] often deem the document retrieval the most heterogeneous task due to the significant surface length and linguistic form differences between its inputs (i.e., the query and the document). Now from the natural language understanding view, we show that answer retrieval is more heterogeneous since it requires quite different understanding abilities on its two inputs.

4 INTERVENTION ANALYSIS

In this section, we further study whether the previous findings on the differences of relevance modeling could actually give us...
some guidelines on model improvement. Inspired by the causal analysis [11], we take the intervention method to study whether some language understanding task could really help to improve the relevance modeling. The core of the intervention method is to take the probing task as the causal factor to interfere the retrieval model, and analyze the change of performances before and after the intervention. Specifically, we first learn the relevance model on each retrieval dataset to obtain the basic results for comparison. Then, we take either features or labels of each intervention factor on the same retrieval dataset to interfere the learning process of the relevance model with other factors hold fixed, and evaluate the performance of intervened models. In the following, we will introduce intervention settings and experimental results in detail.

### 4.1 Intervention Settings

Here, we choose four representative probing tasks as intervention factors, i.e., Keyword, NER, Synonym, and SemArcCls, which is based on the following observations: 1) The Synonym has shown to be consistently improved on all the three retrieval tasks. 2) The SemArcCls has shown to be consistently deceased on all the three retrieval tasks. 3) The Keyword and the NER tasks have obtained distinct improvement on the document retrieval and answer retrieval, respectively. It is worth to note that the intervention process require the retrieval dataset to contain the label of each intervention factor, which would take enormous workloads to obtain the groundtruth label. Recently, the weak labeling method has attracted considerable attention and shown to be beneficial in many NLP tasks [19]. Therefore, we take the finetuned Bert large 5, which has been proved to be effective in all four intervention factors, to generate weak labels for each instance in all three retrieval datasets (i.e., Robust04, MsMarco, and Ubuntu). Then, the label of each intervention factor is used to interfere the learning process of the retrieval model. The details of each intervention method are described as follows:

- **Feature Intervention:** For feature intervention, we take the label of each instance as an additional input to the retrieval model. Specifically, we map the label of each factor (e.g., PER, ORG, LOC in the NER) to embedding space and add the feature embedding to the BERT input embeddings. Thus, the final input embeddings of the retrieval model are the sum of the token embeddings, the segmentation embeddings, the position embeddings, and the feature embeddings. Here, the embedding size of each feature is set to 768 as is in original BERT model.

- **Parameter Intervention:** For parameter intervention, we firstly learn the retrieval model using the label of each intervention factor as the initial parameter, and then finetune the parameters of the model with each retrieval dataset. It is worth to note that we add an additional multi-layer perceptron layer on top of the relevance model to adapt it for each intervention factor. In the experiments, we learn each intervention factor with a small learning rate of $1e^{-5}$, and finetune on the retrieval task with learning rate of $3e^{-5}$.

- **Objective Intervention:** For objective intervention, we jointly learn the intervention factor as well as the retrieval task. For this purpose, we add a task-specific layer on top of the Bert model for each intervention factor. For example, we add a CRF layer on top of the Bert for sequence labeling tasks (i.e., NER), and add a linear layer on top of the Bert for classification tasks (i.e., Keyword, SemArcCls, and Synonym).

The loss function is a weighted sum of the ranking cross-entropy function and factor-specific loss function:

$$Loss = \lambda Loss_{\text{ranking}} + (1 - \lambda) Loss_{\text{factor}},$$

where the $\lambda$ is learned in an end-to-end way.

### 4.2 Results

In this section, we show the intervention results of each intervention factors, including the comparison of different intervention methods and the analysis of different intervention factors.

#### 4.2.1 Intervention Methods Comparison

Here, we compare each intervention method on all the three retrieval tasks based on the intervention factor of **Keyword**. The overall results are summarized in Table 3. Firstly, we can see the that the feature intervention has very little effect on the performances of all the three retrieval tasks. This maybe that the embedding features of each token, which are built on the corresponding intervention factor, are not much effective for the retrieval modeling. Secondly, the objective intervention has significantly decreased the retrieval performances on all the three retrieval tasks with a large margin. The reason may be that the multi-task learning could possibly introduce inductive bias, which would lead to sub-optimal performances on individual tasks [1]. Finally, the parameter intervention has gained significant improvements on the document retrieval task, and dropped with a large margin on the answer retrieval task and the response retrieval task. This is consistent with the previous findings on the probing analysis section, and verifies the importance of keyword recognition in the retrieval modeling of document retrieval. All these results demonstrate that the parameter intervention is more effective than the other two intervention methods.

#### 4.2.2 Intervention Factors Analysis

In this section, we further study whether and how different intervention factors could improve the relevance modeling through the parameter intervention. The intervention results are summarized in Table 4. Firstly, we can see that the NER and **Keyword** have significantly improved the performances of the retrieval model in answer retrieval and document retrieval, respectively. For example, the NER improved the BERT$_{ans}$ with a

| intervention type  | BERT$_{doc}$ | BERT$_{ans}$ | BERT$_{rsp}$ |
|--------------------|-------------|-------------|-------------|
| feature            | 0.459       | 0.367       | 0.817       |
| parameter          | 0.468 (+2%*)| 0.355 (-11.7%) | 0.721 (-3.3%) |
| objective          | 0.402 (-12.4%) | 0.341 (-8.7%) | 0.746 (+7.1%) |
In answer retrieval, different methods are proposed to model the relevance between the answer and the query. The relevance modeling in these three retrieval tasks. Finally, it can be observed that the Synonym has consistently improved the performances of all relevance models. For example, the relative improvements of each retrieval model are 0.83%, 0.5%, and 1.1% on document retrieval, answer retrieval, and response retrieval, respectively. Moreover, it is worthy to note that the labels are automatically generated by an effective model of each intervention factor, which are often somewhat noisy and uncertain. Thus, it would be expected to further enhance the retrieval model if there exists ground-truth labels for each intervention factor. All these results demonstrate that the factors revealed in the probing analysis could really be helpful to the relevance modeling for different retrieval tasks.

5 RELATED WORKS

In this section, we will introduce the works related to our study, including the relevance modeling and the probing analysis.

5.1 The Relevance Modeling

Relevance modeling is a core research problem in information retrieval. During the past decades, researchers have proposed a numerous number of relevance models for different retrieval tasks. In the document retrieval, different kinds of methods have been proposed to measure the relevance between a query and a document [14], including traditional heuristic methods [40, 41], learning to rank methods [4, 21], and neural ranking methods [13]. Firstly, the traditional heuristic methods, such as BM25, and TFIDF, build the heuristic function based on term frequencies, document length, and term importance. Then, the learning to rank models try to learn the ranking function based on machine learning methods on human designed features. Finally, the neural models, which attracted a great attentions in recent years, automatically learn the ranking function as well as the features based on neural networks [33]. In answer retrieval, different methods are proposed to model the relevance relationship between the questions and the answers [13, 27, 29]. In early days, traditional methods focused on the feature engineering, such as lexical, syntactic, and semantic features [29, 58]. Recently, deep learning methods have significantly improved the answer ranking tasks, and become a mainstream method in this task [22, 27, 55]. In response retrieval, the relevance model is designed to evaluate the relevance degree between the utterances and the responses. Early methods was designed with handcrafted templates and heuristic rules. In recent years, neural models have become the mainstream along with the large scale human-human conversation data available [6, 52, 56], and pushed forward the development of the conversation systems.

Though numerous relevance models have been introduced by considering requirements under different retrieval tasks, there has few works try to analyze the relevance modeling under different retrieval applications. To the best of our knowledge, this is the first work to study the relevance modeling in different retrieval tasks based on the empirical analysis.

5.2 The Probing Analysis

Recently, the probing tasks have been widely used to understand the powerful neural models, since the neural models often serve as a black-box in the downstream tasks. The core of the probing methods is to apply the linguistic tasks on the target model to investigate the properties based on the performance of these tasks. A number of works have been proposed to study the linguistic properties of the learned representations over neural models [2, 7, 35]. For example, Belinkov et al. [2] investigated the morphology information through several probing tasks like part-of-speech and semantic tagging on neural MT models. Conneau et al. [7] constructed a set of probing tasks to study the linguistic properties of sentence embedding. There are also several works try to investigate how to design a good probe for the model understanding [53, 60]. For example, Zhang et al. [60] presented experiments to understand how the training sample size and memorization affect the performance of linguistic tasks. Hewitt and Liang [17] investigated the selectivity of probes, and proposed control tasks to study the expressivity of probe tasks and methods.

These studies inspired us to analyze the relevance modeling in different retrieval tasks. However, most of the existing works take the probing tasks directly to investigate the property of the pre-trained language model, we instead use them to compare the focuses of retrieval tasks under different applications.

6 CONCLUSION

In this paper, we present an empirical analysis of the relevance modeling in different retrieval tasks, including document retrieval, answer retrieval, and response retrieval. We propose to use the probing method to investigate the relevance modeling, and introduce 16 probing tasks for the relevance analysis. The results show some interesting findings about the focuses of different retrieval tasks. To further study how to leverage these findings to improve the relevance learning in each retrieval task, we introduce three intervention methods, i.e., feature intervention, parameter intervention, and objective intervention, to interfere existing retrieval models. The intervention results demonstrate that it is able to improve the retrieval models based on the findings on language understanding by carefully designed intervention methods. The analysis of the
relevance modeling is a foundation for designing effective relevance models in real world applications. We believe the way we study the problem (probing & intervention) as well as our findings would be beneficial to the IR community. For future work, we’d like to apply the findings of the probing analysis to improve existing retrieval models. Moreover, we would also try to design new effective retrieval models based on the findings in this work.

7 ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (NSFC) under Grants No. 61902381, 61722211, 61773362, and 61872338, and funded by Beijing Academy of Artificial Intelligence (BAAI) under Grants No. BAAI2019ZD0306 and BAAI2020ZJ0303, the Youth Innovation Promotion Association CAS under Grants No. 20144310, and 2016102, the National Key RD Program of China under Grants No. 2016YQ02D0405, the Lenovo-CAS Joint Lab Youth Scientist Project, the K.C.Wong Education Foundation, and the Foundation and Frontier Research Key Program of Chongqing Science and Technology Commission (No. cstc2017jcyjBX0059).

REFERENCES

[1] Rahaf Aljundi, Pinarjay Chakravarty, and Tinne Tuytelaars. 2017. Expert gate: Lifelong learning with a network of experts. In CVPR. 3366–3375.

[2] Yonatan Belinkov, Nadir Durran, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? arXiv preprint arXiv:1704.03471 (2017).

[3] Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. TACL 7 (2019), 49–72.

[4] Christopher JC Burges. 2010. From random to labyrinthrank to lambdamart: An overview. Learning 11 (2010), 23–58.

[5] Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A Survey on American documentation. Journal of the ACM 7, 5 (2019), 343–469.

[6] Stefano Mizzaro. 1997. Relevance: The whole history. JASIS 48, 9 (1997), 810–832.

[7] G. Salton, A. Wong, and C. S. Yang. 1975. A Vector Space Model for Automatic Indexing. Commun. ACM 18, 11 (Nov. 1975), 613–620.

[8] Stephen Robertson and K. Sparck Jones. 1976. Relevance weighting of search indexing and information retrieval. In Proceedings of the 5th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 22–26.

[9] Rafael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In ACL. Association for Computational Linguistics, 541–550.

[10] Anette Huth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In EMNLP. Association for Computational Linguistics, 216–223.

[11] Jakob Thuesen Joachims. 2006. Training linear SVMs in linear time. In SIGKDD. ACM, 217–226.

[12] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering. In Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28–August 2, 2019, Volume 1: Long Papers, Anna Korhonen, David R. Traum, and Luis Márquez (Eds.). Association for Computational Linguistics, 6086–6096. https://doi.org/10.18653/v1/p19-1612

[13] Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted Supersense Disambiguation of English Prepositions and Possessives. arXiv preprint arXiv:1201.04805 (2019).

[14] Herman Van Willigen and John Larry Kuhn. 1960. On relevance, probabilistic indexing and information retrieval. Journal of the ACM (JACM) 7, 3 (1960), 216–244.

[15] Bhaskar Mitra and Nick Crawell. 2017. Neural Models for Information Retrieval. arXiv: Information Retrieval (2017).

[16] Stefano Mizzaro. 1997. Relevance: The whole history. JASIS 48, 9 (1997), 810–832.

[17] Erik F Sang and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 shared task: Semantic role labeling. In Proceedings of the 4th Conference on Empirical Methods in Natural Language Processing (EMNLP-2000). Association for Computational Linguistics, 39–46.

[18] Erik F Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: A statistical approach to semantic role labeling. In Proceedings of the 6th Conference on Empirical Methods in Natural Language Processing (EMNLP-2003). Association for Computational Linguistics, 462–469.

[19] Matthew E Peters, Mark Neumann, Luke Zettlemoyer, and Wren-tau Yih. 2018. Dissecting contextual word embeddings: Architecture and representation. arXiv preprint arXiv:1808.09949 (2018).

[20] G. Salton, A. Wong, and C. S. Yang. 1975. A Vector Space Model for Automatic Indexing. Commun. ACM 18, 11 (Nov. 1975), 613–620.

[21] Erikk S Fisk and Sahine Buchholz. 2000. Introduction to the CoNLL-2000 shared task: Chunking. arXiv preprint cs/0009008 (2000).

[22] G. Salton. 1989. Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information. McGraw-Hill, Inc, New York.

[23] Stephen Robertson and K. Sparck Jones. 1976. Relevance weighting of search indexing and information retrieval. In Proceedings of the 5th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 22–26.

[24] Stephen Robertson and K. Sparck Jones. 1976. Relevance weighting of search indexing and information retrieval. In Proceedings of the 5th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 22–26.

[25] Stephen Robertson and K. Sparck Jones. 1976. Relevance weighting of search indexing and information retrieval. In Proceedings of the 5th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 22–26.

[26] Stephen Robertson and K. Sparck Jones. 1976. Relevance weighting of search indexing and information retrieval. In Proceedings of the 5th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 22–26.

[27] Stephen Robertson and K. Sparck Jones. 1976. Relevance weighting of search indexing and information retrieval. In Proceedings of the 5th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 22–26.

[28] Stephen Robertson and K. Sparck Jones. 1976. Relevance weighting of search indexing and information retrieval. In Proceedings of the 5th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 22–26.

[29] Stephen Robertson and K. Sparck Jones. 1976. Relevance weighting of search indexing and information retrieval. In Proceedings of the 5th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 22–26.

[30] Stephen Robertson and K. Sparck Jones. 1976. Relevance weighting of search indexing and information retrieval. In Proceedings of the 5th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 22–26.

[31] Stephen Robertson and K. Sparck Jones. 1976. Relevance weighting of search indexing and information retrieval. In Proceedings of the 5th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 22–26.

[32] Stephen Robertson and K. Sparck Jones. 1976. Relevance weighting of search indexing and information retrieval. In Proceedings of the 5th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 22–26.

[33] Stephen Robertson and K. Sparck Jones. 1976. Relevance weighting of search indexing and information retrieval. In Proceedings of the 5th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 22–26.

[34] Stephen Robertson and K. Sparck Jones. 1976. Relevance weighting of search indexing and information retrieval. In Proceedings of the 5th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 22–26.

[35] Stephen Robertson and K. Sparck Jones. 1976. Relevance weighting of search indexing and information retrieval. In Proceedings of the 5th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 22–26.

[36] Stephen Robertson and K. Sparck Jones. 1976. Relevance weighting of search indexing and information retrieval. In Proceedings of the 5th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 22–26.

[37] Stephen Robertson and K. Sparck Jones. 1976. Relevance weighting of search indexing and information retrieval. In Proceedings of the 5th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 22–26.

[38] Stephen Robertson and K. Sparck Jones. 1976. Relevance weighting of search indexing and information retrieval. In Proceedings of the 5th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 22–26.

[39] Stephen Robertson and K. Sparck Jones. 1976. Relevance weighting of search indexing and information retrieval. In Proceedings of the 5th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 22–26.

[40] Stephen Robertson and K. Sparck Jones. 1976. Relevance weighting of search indexing and information retrieval. In Proceedings of the 5th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 22–26.

[41] Stephen Robertson and K. Sparck Jones. 1976. Relevance weighting of search indexing and information retrieval. In Proceedings of the 5th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 22–26.
[47] Natalia Silveira, Timothy Dozat, Marie-Catherine De Marneffe, Samuel R Bowman, Miriam Connor, John Bauer, and Christopher D Manning. 2014. A Gold Standard Dependency Corpus for English. In LREC. 2897–2904.

[48] Craig Silverstein, Hannes Marais, Monika Henzinger, and Michael Moricz. 1999. Analysis of a Very Large Web Search Engine Query Log. SIGIR Forum 33, 1 (Sept. 1999), 6–12. https://doi.org/10.1145/331403.331405

[49] Antoine Tixier, Fragkiskos Malliaros, and Michalis Vazirgiannis. 2016. A graph degeneracy-based approach to keyword extraction. In EMNLP. 1860–1870.

[50] Ellen M Voorhees. 2006. Overview of the TREC 2005 Robust Retrieval Track. In Text Retrieval Conference (TREC).

[51] Baowin Wang. 2018. Disconnected recurrent neural networks for text categorization. In ACL. 2311–2320.

[52] Taesun Whang, Dongyub Lee, Chanhee Lee, Kisu Yang, Dongsuk Oh, and HeuSeok Lim. 2019. Domain Adaptive Training BERT for Response Selection. arXiv preprint arXiv:1908.04812 (2019).

[53] Zhiyong Wu, Yun Chen, Ben Kao, and Qun Liu. 2020. Perturbed Masking: Parameter-free Probing for Analyzing and Interpreting BERT. In Proceedings of the 58th ACL, Online, July 5-10, 2020. 4166–4176.

[54] Rui Yan, Yiping Song, and Hua Wu. 2016. Learning to Respond with Deep Neural Networks for Retrieval-Based Human-Computer Conversation System. (2016), 55–64.

[55] Liu Yang, Qingyao Ai, Jiafeng Guo, and W Bruce Croft. 2016. aNMM: Ranking short answer texts with attention-based neural matching model. In Proceedings of the 25th ACM international on conference on information and knowledge management. 287–296.

[56] Liu Yang, Minghui Qiu, Chen Qu, Jiafeng Guo, Yongfeng Zhang, W. Bruce Croft, Jun Huang, and Haqing Chen. 2018. Response Ranking with Deep Matching Networks and External Knowledge in Information-seeking Conversation Systems. In The 41st International ACM SIGIR, Ann Arbor, USA, July 08-12, 2018. ACM. 245–254. https://doi.org/10.1145/3209978.3210011

[57] Michihiro Yasunaga, Jungo Kasai, and Dragomir Radev. 2017. Robust multilingual part-of-speech tagging via adversarial training. arXiv preprint arXiv:1711.04903 (2017).

[58] Wentao Yih, Mingwei Chang, Christopher Meek, and Andrzej Pastorasiak. 2013. Question Answering Using Enhanced Lexical Semantic Models. (2013), 1744–1753.

[59] Hamed Zamani, Mostafa Dehghani, W Bruce Croft, Erik Learned-Miller, and Jaap Kamps. 2018. From neural re-ranking to neural ranking: Learning a sparse representation for inverted indexing. In CIRI. ACM. 497–506.

[60] Kelly W. Zhang and Samuel R. Bowman. 2018. Language Modeling Teaches You More Syntax than Translation Does: Lessons Learned Through Auxiliary Task Analysis. CoRR abs/1809.10040 (2018). arXiv:1809.10040 http://arxiv.org/abs/1809.10040

[61] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In NIPS. 649–657.

[62] Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase Adversaries from Word Scrambling. arXiv preprint arXiv:1904.01130 (2019).
A RETRIEVAL TASK SETTING

To learn the relevance model for each retrieval task, here, we following existing works to apply the corresponding ranking loss on each datasets. Specifically, for Robust04 dataset, we utilize the pairwise ranking loss (i.e., hinge loss) [8] to train the retrieval model, i.e., given a triple \((s, t^+, t^-)\), where \(t^+\) is ranked higher than \(t^-\) with respect to a query \(s\), the loss function is defined as:

\[
L(s, T^+; T^-, \theta) = \sum_{t^+ \in T^+} \max(0, 1 - f(s, t^+) + s(s, t^-)),
\]

where \(f(s, t)\) denotes the relevance score for pair \((s, t)\), and \(\theta\) includes all learnable parameters in the ranking model. For MsMarco and Ubuntu dataset, we take the cross entropy loss [31, 52] to train the retrieval model,

\[
L(s, T^+, T^-, \theta) = -\sum_{j \in T^+} \log(f(s, t_j)) - \sum_{j \in T^-} \log(f(s, t_j)),
\]

where \(T^+\) and \(T^-\) denote the positive and negative answers/responses with respect to question/utterance \(s\). The optimization is relatively straightforward with standard backpropagation. We apply stochastic gradient decent method Adam with learning rate warmup over the first 10% training steps, and linear decay of the learning rate for BERT layers, fixed learning rate of 0.001 for task-specific layer which always be a linear classification layer. We use a dropout probability of 0.1 on all layers. Since the length of text inputs in each retrieval tasks differ significantly wit each other, we thus tailor the input length for each dataset accordingly. Specially, for Robust04 dataset, the maximum sequence length is set to 512 where the length of left and right input is set to 30 and 480, respectively. For MSMARCO dataset, the maximum sequence length is set to 230, where the length of left and right input is set to 30 and 200, respectively. For Ubuntu dataset, the maximum sequence length is set to 300, where the length of left and right input is set to 256 and 44, respectively.

B REFERENCES TO STATE-OF-THE-ART MODELS ON THREE RETRIEVAL TASKS

| Task            | Previous state of the art (without pretraining) | BERT       |
|-----------------|-----------------------------------------------|------------|
| Robust04        | 43.1 (Zamani et al., 2018)                     | 46.9 (Dai and Callan, 2019) |
| MsMarco         | 27.1 (Dai et al., 2018)                        | 35.8 (Nogueira and Cho, 2019) |
| Ubuntu          | 79.6 (Chen and Wang, 2019)                     | 81.7 (Whang et al., 2019)   |

Table 5: A Comparison of Performance of prior state of the art models and BERT.

C PROBING SETTINGS

The statistics of the dataset used in each probing tasks are listed in Table 6. For all the probing experiments, we follow existing works to use linear probe on all these tasks as it has been proved to have better selectivity [16]. Specifically, we add a linear layer on top of each layer in the Bert model as the prediction layer of each probing task. We follow Liu et al. 2019’s work and take the contextual-repr-analysis toolkit for the probing experiments. This toolkit is implemented under the AllenNLP (Gardner et al., 2017) framework. The description of the probe datasets are listed in the Table 6. Note that we build two novel probing tasks, i.e., Synonym and Polysemy, to directly evaluate the semantic understanding of word pairs. For performance evaluation, we take the F1 metric for Chunk, NER, GED and Keyword, while the rest are based on the Acc metric [24]. It is worth to note that the vocabulary size of each retrieval dataset differs significantly, which would impact the performance of downstream probing tasks. To make a fair comparison, we remove the instances where the tokens is out of the target vocabulary (i.e., the vocabulary of retrieval datasets) from the probe datasets. All the probing tasks are tuned with Adam with batch size of 80, using a learning rate of 0.001 for maximum number of 50 epochs, using early stopping with a patience of 3.

| Probing Tasks          | Train | Dev  | Test | Metric |
|------------------------|-------|------|------|--------|
| Chunk                  | 6k    | 1.7k | 1.7k | F1     |
| POS                    | 11k   | 1.8k | 1.8k | Acc    |
| NER                    | 12k   | 2.7k | 2.9k | F1     |
| GED                    | 27k   | 2.1k | 2.7k | F1     |
| SynArcPred             | 11k   | 1.8k | 1.8k | Acc    |
| SynArcCls              | 11k   | 1.8k | 1.8k | Acc    |
| Word Scramble          | 49k   | 8k   | 8k   | Acc    |
| PS-fxn                 | 2.4k  | 0.5k | 0.5k | Acc    |
| PS-role                | 2.4k  | 0.5k | 0.5k | Acc    |
| CorefArcPred           | 2.2k  | 0.2k | 0.2k | Acc    |
| SemArcPred             | 25k   | 2.5k | 2.5k | Acc    |
| SemArcCls              | 25k   | 2.5k | 2.5k | Acc    |
| Synonym                | -     | -    | 10k  | Acc    |
| Polysemy               | -     | -    | 7.6k | Acc    |
| Keyword Extract        | 109k  | 50k  | 50k  | F1     |
| Topic Classification   | 100k  | 20k  | 20k  | Acc    |

Table 6: Statistics of the datasets of each probing task.

D REFERENCES TO STATE-OF-THE-ART TASK-SPECIFIC MODELS (WITHOUT PRETRAINING)

| Task        | Previous state of the art (without pretraining) |
|-------------|-----------------------------------------------|
| POS         | 95.82 (Yasunaga et al., 2017)                  |
| Chunk       | 95.77 (Hashimoto et al., 2016)                 |
| NER         | 91.38 (Hashimoto et al., 2016)                 |
| GED         | 39.83 (Rei and Søgaard, 2019)                  |
| PS-Role     | 66.89 (Schneider et al., 2018)                 |
| PS-Fxn      | 78.29 (Schneider et al., 2018)                 |
| Keyword Extract | 56.09 (Tixier et al., 2016)               |
| Topic Classification | 76.26 (Wang, 2018)                           |

Table 7: Performance of prior state of the art models (without pretraining) for each probe task.