ANSWER SIMILARITY GROUPING AND DIVERSIFICATION IN QUESTION ANSWERING SYSTEMS

ANSWER SIMILARITY GROUPING AND DIVERSIFICATION IN QUESTION ANSWERING SYSTEMS

A Dissertation Presented

by

LAKSHMI NAIR VIKRAMAN

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

September 2022

Robert and Donna Manning College of
Information and Computer Sciences

ANSWER SIMILARITY GROUPING AND DIVERSIFICATION IN QUESTION ANSWERING SYSTEMS

A Dissertation Presented

by

LAKSHMI NAIR VIKRAMAN

Approved as to style and content by:

W. Bruce Croft, Chair

James Allan, Member

David Jensen, Member

Rajesh Bhatt, Member

James Allan, Chair of the Faculty
Robert and Donna Manning College of
Information and Computer Sciences

ACKNOWLEDGMENTS

I would like to take this opportunity to express my gratitude to everyone who helped me during the PhD journey.

First, I would like to thank my committee members, W. Bruce Croft, James Allan, David Jensen and Rajesh Bhatt for their feedback during various stages of the journey. Bruce introduced me to the initial core QA task which formed the basis of the thesis, while the comments and questions raised by James and David helped me look more deeply at the work to better understand the reasoning behind the model performance. I would also like to thank my previous advisor, Andrew McCallum for accepting me into the PhD program, thereby providing a stepping stone to rise higher.

I would like to thank all the labmates (in CIIR and IESL) with whom I interacted during my time at Amherst as well as the staff, including Dan Parker, Jean Joyce, Stephen Harding and others. Finally, I would also like to thank current and earlier CICS department staff members, Eileen Hamel, Kyle Skemmer and Leeanne M. Leclerc, who provided significant support and smoothed many tough spots, making the process easier to navigate.

This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF IIS-1715095. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

ABSTRACT

ANSWER SIMILARITY GROUPING AND DIVERSIFICATION IN QUESTION ANSWERING SYSTEMS

SEPTEMBER 2022

LAKSHMI NAIR VIKRAMAN

B.Tech., UNIVERSITY OF KERALA

M.S., COLUMBIA UNIVERSITY

PhD., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor W. Bruce Croft

The rise in popularity of mobile and voice search has led to a shift in IR from document to passage retrieval for non-factoid questions. Various datasets such as MSMarco, as well as efficient retrieval models have been developed to identify single best answer passages for this task. However, such models do not specifically address questions which could have multiple or alternative answers. In this dissertation, we focus on this new research area that involves studying answer passage relationships and how this could be applied to passage retrieval tasks.

We first create a high quality dataset for the answer passage similarity task in the context of question answering. Manual annotation of passage pairs is performed to set the similarity labels, from which answer group information is automatically generated. We next investigate different types of representations, which could be used to create effective clusters. We experiment with various unsupervised representations

and show that distributional representations outperform term based representations for this task. Next, weak supervision is leveraged to further improve the cluster modeling performance. We use BERT as the underlying model for training and show the relative performance of various weak signals such as GloVe and term-based Language Modeling for this task. In order to apply these clusters to the answer passage retrieval task for multi-answer questions, we use a modified version of the Maximal Marginal Relevance (MMR) diversification model. We demonstrate that answers retrieved using this model are more diverse i.e, cover more answer types with low redundancy as well as maximize relevance, with respect to the baselines. So far, we used passage clustering as a means to identify answer groups corresponding to a question and apply them in a question answering task. We extend this a step further by looking at related questions within a conversation. For this purpose, we expand the definition of Reciprocal Rank Fusion (RRF) and use this to identify pertinent history passages for such questions. Updated question rewrites generated using these passages are then used to improve the conversational search task.

In addition to being the first work that looks at answer relationships, our specific contributions can be summarized as follows: (1) Creation of new datasets with passage similarity and answer type information; (2) Effective passage similarity clustering models using unsupervised representations and weak supervision methods; (3) Applying the passage similarity/clustering information to diversification framework; (4) Identifying good response history candidates using answer passage clustering for the conversational search task.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

Answer Passage Retrieval is an emerging topic in Information Retrieval (IR), where the goal is to display relevant passage texts in response to the information needs expressed by a user. This is especially suited to applications such as mobile and voice search, where the display space or response bandwidth is limited, and the short answer passages provide a good alternative to displaying a list of documents. Personal Assistants such as Siri and Cortana are examples of such applications, which are used by users to complete tasks or satisfy more complex information needs. Users interact with such systems either by issuing single questions (single turn question answering) or via a series of questions covering single or related topics (multi-turn conversational search). In this dissertation, we primarily focus on the question answering task, while also demonstrating how techniques effective for this could be useful in a conversational search task setting.

The primary aim of a question answering system in IR is to take a question as input, search a passage collection for corresponding answers and return the most relevant answer passages. Based on the type of information need, users could issue different type of questions. For example, some questions are focused and specific and could be answered by short entity level answers, while others are more open-ended and have answers spanning multiple sentences. The first type of questions are called factoid questions and the second are non-factoid questions. Examples of different question types are given in Table 1.1. The factoid question is from the SQuAD dataset, proposed by Rajpurkar et al. [80] and the non-factoid question is from the

ANTIQUE dataset proposed by Hashemi et al. [30]. Factoid questions have been studied extensively in other areas such as NLP [36, 80, 79, 45], however, non-factoid QA is a relatively new area and is less explored due to its relative complexity. In this dissertation, we focus on the second type, i.e. non-factoid questions.

Table 1.1 Types of Questions

Type	Question	Answer
Factoid	What is the Dutch word for the Amazon rainforest?	Amazoneregenwoud
Non-factoid	What is innate immunity?	Immunity that occurs naturally as a result of a person's genetic constitution or physiology and does not arise from a previous infection or vaccination. Also called genetic immunity, inherent immunity, native immunity, natural immunity, nonspecific immunity.

To stimulate research in this area, the IR research community has built several datasets [38, 13, 62, 16, 30] and explored different types of models including complex neural models [94, 13, 14, 15, 64, 66]. These generally use open-retrieval settings, where a term-based baseline is re-ranked using an objective function, optimized to return a correct answer at the top of the ranked list. However, these datasets and models do not explicitly address questions which could have multiple answers covering different answer types. Hence, we focus on studying non-factoid question answering, where answers to questions could cover different types.

We approach this problem by first studying how answer passage clustering could be an effective strategy to group answers into these answer types. We then present how this could be used within a diversification framework to display answers and in a conversational setting to improve its overall effectiveness.

1.1 Passage Retrieval Datasets

The first step to enable this research is the creation of high-quality datasets. A number of datasets [38, 13, 62, 16, 30] for non-factoid question answering exists which capture the relevance of answer passages with respect to the questions. However, to facilitate this research, information with respect to answer groups must also be available. A potential source of acquiring such labelled data is from the TREC Web Track's diversity datasets [12, 88] related to the Search Result Diversification task [111, 9, 98, 83, 23, 32]. However, these document diversification datasets cover query facets which is different from answer types corresponding to descriptive questions. We present an example to demonstrate how they differ in Chapter 5. To potentially train large models with answer group information, datasets employed in similarity tasks could be useful. Sentence similarity tasks have been studied in NLP including textual entailment [8, 97, 63] and sentence paraphrasing [37, 40]. However, these datasets cover sentence level information and do not extend to passage level. Therefore, in this dissertation, we study the task using unsupervised and weakly supervised methods and propose the creation of a new test dataset for evaluation purposes. We perform manual annotation to create the NFPassageQA_Sim dataset covering passage pair similarity annotations for 128 questions and then generate NFPassageQA_Div dataset with answer group information.

1.2 Answer Passage Clustering

To group answers into answer types, the initial step is to perform answer grouping or clustering based on similarity measures. For instance, for the question, `How can we get rid of mice?`, there could be answers corresponding to different answer types `Animal Control`, `Traps` etc. There could also be answers which cover both these types. In order to find answer groups, the similarity metric must assign a higher value to answers within a group as opposed to answers outside it. A classic

IR task related to this is Cluster-based Retrieval, where clusters of documents are retrieved in response to a query [57, 96, 58, 44, 43, 78, 86]. However, one of the main limitations of these models is that they group documents/passages using term based Language Modeling (LM) strategies, which use term distribution information for estimating similarities and do not capture additional semantic information. In this dissertation, we propose to explore the effectiveness of this task using pre-trained distributed representations such as GloVe [68], which capture term level semantic information and BERT [25], which captures semantic information with longer contexts such as sentences and passages, in addition to the classic term based models.

One of the drawbacks of the pre-trained representations is the lack of task-specific information, which could be captured by training models on large training sets. In the absence of training data, another option is to use weak supervision strategies to train models. Weak supervision provides an alternative to human labelled data by leveraging other easily available sources. There is existing work in IR where weak supervision methods using BM25 ranking as weak signals have been shown to be effective for document ranking [24]. More recently, Xu et al. [100] demonstrated the application of weak supervision to passage retrieval tasks where a relatively small training set was used to fine-tune BERT for this task. In this dissertation, we propose a similar weak supervision strategy for answer passage clustering using three sources of weak labels: the Language Model (LM), GloVe [68], and pre-trained BERT [25]. We train a BERT model using different objective functions and learn a good similarity function, which is helpful for grouping similar answers together.

1.3 Answer Passage Diversification

In order to display answers to multi-answer questions, a mechanism to rank them on the basis of the answer types would be useful. For example, in case of the question, `How can we get rid of mice?`, with two potential answer types: `Animal Control`

and `Traps`, ideally the top answers should cover both types with minimum redundancy. Retrieval models optimized to maximize relevance alone, may not achieve this and hence we adopt a diversification framework. Search Result Diversification [111, 9, 98, 83, 23, 32] is a well-studied task in IR, where the end goal is to diversify documents with respect to various query facets or subtopics. Since the questions in the question answering task are well-defined, and diversification in this case must be performed on the basis of answer types, we study how the diversification algorithm could be extended to accommodate this. Broadly, Search Result Diversification models could be categorized into two types: Implicit and Explicit. The implicit models assume that each document represents its own topic and diversifies based on document similarity [111, 9, 98]. On the other hand, the explicit models uses query topics to explicitly diversify the result set [83, 23, 32]. Since, we approach the task using answer similarity clustering models, implicit methods provide a more natural solution. In this thesis, we introduce a modified version of the implicit MMR model, which incorporates the answer passage clusters into a MMR framework and creates a diversified ranked list for displaying answers effectively.

1.4 Passage Retrieval in Multi Turn Conversational Search

Users interact with systems to get their information needs met. An example of such an interaction is with voice agents such as Alexa, Siri, Cortana etc., where one form of exchange takes the form of multi-turn question answering, with users asking follow-up questions or questions related to similar topics. In such a system, the conversational agent may need information from previous queries issued by the user during the conversation, to fully comprehend the current query. In most cases, the previous query turn provides the most relevant information, however, the users can also refer to earlier turns.

A related task is answer passage retrieval in a conversational setting, where answers are retrieved for each question. This task is the main objective of the TREC Conversational Assistance Track (CAsT) [21, 20] initiative. These questions tend to be non-factoid in nature, with potentially multiple answers, each covering multiple sentences. The complexity of the task, along with limitations of training data makes this difficult to model in a fully supervised setting. Many few-shot and weak supervision models [105, 104] have been proposed, which rely on query reformulations to rewrite the queries and then use them in an ad-hoc search scenario. The query reformulations could be in the form of query expansion models [56, 93] or query rewriting models [76, 104]. In this dissertation, we employ the query rewriting approach for the passage retrieval task.

History queries and responses have been utilized to improve conversational question answering results in a Machine Comprehension setting [11, 82, 33]. Many of these models focused on using the immediate preceding turn to model history. More recently, Chen et al. [74, 72, 75] extended the model to include any of the prior history turns using BERT based attention models. However, such models have not been studied in a passage retrieval scenario, primarily due to lack of training data. Many existing models use the entire query history to rewrite the query at the current turn. Another potential source of information for generating good rewrites are history responses. Since the system maintains all utterances and agent responses, a combination of both could generate better rewrites. However, since queries within a conversation are related, instead of using the top response generated by a conversational agent for a previous (or history) query, we could identify other potential responses based on the inter-relationships between the queries. In this dissertation, we study how answer passage clusters created from passages retrieved using baseline query rewrites could be helpful in generating better rewrites using automatically generated response candidates, to improve passage retrieval in a conversational setting.

1.5 Contributions

We summarize our contributions as follows:

- For studying answer passage relationships, we employ manual annotation to create a high quality test set with similarity labels for answer passage pairs (NFPassageQA_Sim). We describe the annotation process in detail and also demonstrate how to automatically use these labels to generate a dataset consisting of answer type information (NFPassageQA_Div). This is the first dataset (to the best of our knowledge) consisting of answer type information for the answer passage retrieval task generated from manually labelled data. Since, any direct annotation effort would require the presence of answer type information, this work demonstrates how the same results could be obtained indirectly.

- We study different types of representations, both unsupervised and weakly supervised and demonstrate experimentally the type of settings which would be effective for the passage similarity task, evaluated using the NFPassageQA_Sim dataset. For the unsupervised setting, we study this with respect to Language Modeling (LM), GloVe, BERT signals and its various combinations and show that GloVe in combination with BERT does the best. In terms of the weakly supervised setting, we experiment with the LM, GloVe, BERT and their combinations as the weak signal and show that GloVe trained using a BERT model performs the best. This indicates that in general distributed representations outperform term-based ones for the similarity task. It also shows that GloVe representations adds useful information to BERT, which explains its effectiveness in both unsupervised and weakly supervised settings.

- We apply passage clusters created using unsupervised and weakly supervised signals and demonstrate how they could be used to improve answer passage ranking in a non-factoid QA setup. We show how the MMR diversification model could be

extended to include passage clusters of top ranking answers retrieved via a baseline model. Experimental results demonstrate that this setting improves the answer ranking quality with respect to both relevance and diversity to generate higher quality answers for multi-answer questions while evaluated using the dataset NF-PassageQA_Div. Further analysis on the results reveal that the improved performance can be attributed to the effectiveness of the passage clusters in identifying more relevant passages and adding them to the final diversified list.

- We also apply passage clustering for improving the passage retrieval task in a conversational setting. We demonstrate that the passage clusters created from inter-related queries in a conversation can be used to identify potential history passage candidates by extending the RRF (Reciprocal Ranking Fusion) method. We identify the new candidate responses for each query and show how these could be used to improve query rewrites by incorporating information from the history passages using a standard rewriter. The updated query rewrites could then used to retrieve better answer candidates in a passage retrieval task. This is the first effort (to our knowledge) at identifying alternative history passage candidates in a fully unsupervised setting. One of the key advantage of using this method is the identification of candidates which maintains the stability of the results in comparison with other baselines. The analysis reveals that adding these response candidates in general maintains the final performance either at or above that of the initial query rewrites.

1.6 Outline

The central focus of this thesis is on creating effective answer passage similarity clusters in a non-factoid QA setting and using these in other tasks such as diversification and conversational search. The rest of the dissertation is organized as follows. Chapter 2 covers Related Work and Background with respect to the underlying Sim-

ilarity task and other related areas. The details of data annotation and generation of answer similarity/type datasets, are covered in Chapter 3. The detailed study of comparative effectiveness of various representations for answer passage similarity task is explained in Chapter 4. Chapter 5 covers how the similarity models are applied to diversity model to generate better answer rankings. Application of passage clustering to conversational search task is described in Chapter 6. Finally, we summarize this dissertation and discuss potential future work in this area.

CHAPTER 2

BACKGROUND AND RELATED WORK

This dissertation is related to several areas of research, namely Question Answering Datasets, Clustering and Representation Learning, Diversification and Conversational Search.

2.1 Question Answering Datasets

Significant work has been done in recent years in the area of Question Answering in both Natural Language Processing (NLP) and IR domains. In IR, this has led to a shift in focus from short queries to longer, more specific questions. Based on the type of answers returned by such questions, they can be categorized as factoid and non-factoid questions. While factoid questions have short entity level answers, non-factoid questions have descriptive answers spanning multiple sentences.

One of the most significant factoid QA datasets is SQuAD [80], introduced by Rajpurkar et al. This dataset was collected in a machine comprehension setting using manual crowdsourcing. The crowdworkers created questions from paragraphs in Wikipedia articles and then highlighted the corresponding answer spans within them. A subsequent version, SQuADRUn[79] was also released, which includes unanswerable questions with plausible but incorrect answers, to account for cases where a correct answer may not be present for a question. Another machine comprehension based QA dataset is the TriviaQA [36] dataset, where the questions were collected from trivia websites with supporting documents gathered independently from Wikipedia or by using Web Search results. In contrast to SQuAD, the questions in TriviaQA

are more challenging, with higher syntactic and lexical variance. Google Natural Questions dataset [46] introduced by Kwiatkowski et al. was also built on the machine comprehension principle, except that this cover both short and a longer version of answers for each question.

One of the first efforts in non-factoid QA dataset creation was by Keikha et al. [38] who introduced the WebAP dataset. This dataset was created from the Gov2 collection and 82 TREC queries. For each of these queries, top 50 documents were first retrieved using the SDM model and the passages from the relevant documents based on the TREC judgements were then assigned one of the 4 relevance judgements. Later, Cohen et al. [13] introduced a relatively larger dataset nfL6, which consists of automatically filtered data from the Yahoo Webscope L6 collection. Since this data was automatically generated without manual intervention, it suffers from incomplete judgments. ANTIQUE [30] is a more recent initiative, created from nfL6, which overcomes the incomplete judgement issue to a large extent by using human annotation and creating a more wide range of potential candidate answers for users to judge. Similar to WebAP, the answers are assigned one of the 4 graded relevance judgements with respect to the questions. Another dataset created using large-scale manual annotation in a Machine Comprehension setting is MSMarco [62]. This was released by Microsoft and consists of questions sampled from Bing's query logs. Relevant documents were retrieved for these questions using Bing's web index and passages from these documents shown to the annotators, who then used them to create natural language answers. Since the crowdsourcing was restricted to passages within top retrieved documents, this could potentially leave out other relevant passages and the annotation is therefore incomplete. WikiPassageQA [16] is another dataset created in a Machine Comprehension setting similar to the SQuAD dataset. Here, human annotators created non-factoid questions from given Wikipedia articles and indicated corresponding answer spans within them. A new dataset which is also created in a

machine comprehension setting is the NLQuAD[89], which identifies long descriptive answers. It contrasts with the machine comprehension datasets described earlier in terms of the source of the supporting documents. While most previous work used Wikipedia, they used BBC news articles to create the dataset.

All the datasets described above capture relevance information of answers with respect to questions. In multi-answer datasets such as ANTIQUE, MS-MARCO etc. it is possible to have answers of different types. For eg: The question, "`How do I treat diabetes?`" could have answer types such as "`Diet`", "`medication`" etc.. However, none of the datasets described so far capture this information. YahooL29 [65], is a CQA dataset which contain answer types corresponding to questions. However, this is limited to questions from Health domain of the Yahoo CQA forum and was created using term clusters. In Chapter 3, we describe the creation of a new evaluation dataset, which overcomes these limitations. The new dataset is open-domain and use standard human annotation to capture answer pair similarities, which are then used to generate answer type information.

2.2 Clustering Models

2.2.1 Clustering models in IR

The cluster hypothesis [34] states that relevant documents which satisfy an information need tend to cluster together. This hypothesis triggered research in the area of Cluster-based retrieval models, where ranked list of documents are retrieved based on the cluster of documents they come from, as opposed to traditional retrieval models where the documents are retrieved based only on the query. Various clustering schemes with retrieval have been studied in IR. Leuski [50] proposed the first study which demonstrated the effectiveness of using clusters created using various hierarchical agglomerative clustering schemes for displaying relevant information. He showed that selecting documents based on a function which interpolates similarity to known

relevant and non-relevant documents can create clusters which can be more effective than traditional ranking approach. Tombros et al. [90] applied hierarchical clustering in a query-specific manner on the top n documents and showed that these clusters performed better than the ones created using the entire collection for the retrieval task. They also inferred that for a query, an optimal cluster exists, which if retrieved would outperform the document retrieval models. However, these conclusions were based on studies conducted by finding the ideal cluster based on available relevance information.

Various methods were introduced to identify and use the optimal clusters to improve retrieval performance. In one type of approach, documents within a higher-ranked cluster retrieved with respect to a query were considered more relevant than documents with respect to the lower-ranked ones. In another type of approach, the clusters were used as a part of document smoothing. Liu et al. [57] proposed two different models based on these two approaches. In the first case, cluster language models instead of document language models were used, and in the second, the document language model was smoothed based on the cluster to which it belonged as well as the collection. They also experimented with static clustering (using K-means) and query-specific clustering and found the static clustering to work well with the smoothed version. The smoothed version was also found to work better than ranking clusters directly to generate document ranked lists and also performed better than standard document retrieval models for many datasets. Wei et al. [96] extended the smoothed version by replacing K-means with LDA (Latent Dirichlet Allocation) to generate the clusters and found this to outperform the K-means version. Liu et al. [58] studied the cluster ranking approach using various cluster representations such as concatenation, centroid etc using query-specific clustering with kNN. They showed that the geometric mean of the document-query match values performed the best with respect to other representations to retrieve the optimal cluster for a query. Us-

ing kNN-based clusters for cluster-based smoothing models was also further studied by Kurland [42], where he demonstrated the relative efficacy of overlapping query-specific kNN clusters in comparison with clustering schemes such as K-means and hierarchical agglomerative clustering.

Instead of using clusters to smooth document models or retrieving documents using only representations for clusters as a whole, Kurland et al. [43] proposed a model which linearly interpolated the cluster score with respect to the query and the scores contributed by the constituent documents within the cluster in a language modeling framework. Raiber et al. [78] added additional cluster-relevance information such as inter-document similarities, query-document similarity within cluster etc. using MRFs (Markov Random Field). Sheetrit et al. [87] showed that the cluster hypothesis applies not only to documents as demonstrated by the earlier work, but also to passages with more focused information. Another work by Sheetrit et al. [86] integrated this to a Learning to Rank framework. In this dissertation, we adopt the same underlying architecture used in the traditional cluster-based models, but instead of using only term-based information such as LM, we also conduct our experiments with representations which capture semantic information.

A related area of interest is the similarity models in NLP. However, their focus is on sentence pair tasks such as textual entailment [8, 97] or paraphrasing [67], which relates to studying similarity between short phrases. Fine-tuned BERT models have been shown to perform very well for many of these tasks [25]. More recent work demonstrates the advantages of applying supplementary training [70] and multi task learning [59] on BERT for the similarity task. Most of the models are trained on large scale datasets and their effectiveness has been demonstrated using fully supervised models. Besides that, almost all the models studied in NLP are based on sentence level or phrase level information, while we study the similarity at passage level, which is a more complex task.

2.2.2 Text Representations

In general, earlier models in IR encoded term frequency information to represent documents. Such models include language modeling (LM), where documents are represented by interpolating Maximum Likelihood and Dirichlet estimates of term counts. Tf-idf is another alternative, where each document is represented by a sparse vector, with each dimension encoding a combination of term frequency (tf) and document frequency (idf) information. Last few years have seen the emergence of pre-trained models such as GloVe [68] and word2vec [61], trained on large easily available unlabelled data such as Wikipedia, to generate dense embeddings for terms. These are based on the basic idea that terms occurring in similar contexts would be more correlated and should have similar representations. The word2vec [61] model is trained on an objective using a classification loss function to encode words to predict surrounding word contexts, while the GloVe [68] model encodes the term co-occurrence more explicitly using a regression loss. Longer text representations such as sentences, documents etc could be created from such term vectors by combining them via concatenation, averaging etc. Other direct methods of generating vectors for such long texts were also investigated. Kiros et al. [39] extended the objective of the word2vec model to generate sentence representations, where a sentence predicts the sentences around it and Le et al. [49] added an a paragraph vector to the word2vec model to capture the topic of the paragraph. Later, Peters et al. [69] introduced the ELMo term representations learned using a deep bidirectional language model with entire sentence as context. More recently, Devlin et al. [25] proposed BERT, which encodes word and sentence/passage level vectors using left and right contexts across all the layers of a deep neural model and encode additional information using position, segment and token embeddings. We use the traditional LM, GloVe and BERT based representations in our experiments for answer passage similarity.

The text representations described above were created using language model based training, where a part of the predicted text was hidden during the process. Though they capture general syntactic and semantic information, more task-specific information could enhance the effectiveness of such representations. Training supervised models using task-specific labelled data provides a means to achieve this. However, for tasks which do not have large-scale labelled training data, an alternative is weak supervision models. They use noisy data labels instead of ground truth labels with the same objective as the supervised models. This has been used effectively in many Information Retrieval tasks. Dehghani et al. [24] demonstrated the effectiveness of using BM25 rankings as weak supervision signal for the task of document ranking. MacAvaney et al. [60] extended this model by using content based sources in addition to ranking sources and proposed a filtering mechanism to overcome domain mismatch. Weak Supervision for Query Performance Prediction (QPP) model was introduced by Zamani et al. [108], where a joint model was trained to combine multiple complementary signals to contribute to the end task. Xu et al. [100] applied weak supervision to passage ranking task by using BERT as the basic framework for training. They generated weak labels by combining multiple weak signals using majority voting and also by learning a simple generative model to predict the labels [81]. The theoretical basis behind the effectiveness of such models was shown by Zamani et al. [107]. Zhao et al. [110] applied this to ad-hoc cross-lingual Information Retrieval task for low resource languages such as Swahili and Tagalog. In this dissertation, we explore how weak supervision can be used to improve the performance of the passage clustering task.

2.3 Diversification Models

The focus of ad-hoc document retrieval tasks is to retrieve and display highly relevant documents with respect to a query, without taking into account various

aspects (also referred to as facets or topics), these queries could possibly cover. For example, the query "apple", may refer to the "fruit" or the "company", and it would be beneficial if the top results cover both these diverse aspects.

Search Result Diversification models were developed to overcome this limitation, by scoring documents based on both relevance and diversity information. This has also been studied under TREC Web Track's diversity tasks ([12, 88]). Broadly, they can be categorised into two types. The first is the implicit model, where the diversification is performed based on the assumption that every document is its own topic. The earliest model of this type is the Maximal Marginal Relevance (MMR) [9] model, where each document is represented as a sparse vector and a document in the ranked list is selected based on its similarity with respect to the already selected documents. Zhu et al. [111] proposed a Relational Learning-to-Rank (R-LTR) framework, which extended the Learning-to-Rank model to include relationships between documents using various features such as text diversity, subtopic diversity, anchor text diversity etc. Further, Xia et al. [98] adopted the R-LTR features and developed a diversity learning algorithm using a Perceptron framework.

In contrast with implicit models, the explicit modeling approach ranks documents based on query topic coverage. Santos et al. [83] described this succinctly as "Given an initial ranking R for a query q, find a re-ranking S that has the maximum coverage and the minimum redundancy with respect to the different aspects underlying q". Using this basic idea, they proposed a probabilistic model xQUAD, where a set of sub-queries potentially covering various aspects of a query could be used for diversification. In this work, the sub-queries were derived from query reformulations provided by a commercial search engine. The coverage of the documents with respect to these sub-queries and aspect coverage in already selected documents were used to generate the final diversified list. Dang and Croft [22] extended this model to use topic terms generated using DSPApprox algorithm [47, 48] instead of query reformulations. They

also proposed a new model PM-2 [23], where documents were selected proportional to the topic popularity. They applied the Sainte-Laguë method to maintain the proportionality, using aspects covered in Santos et al [83] and aspect judgements from TREC. Later, Dang and Croft [22] successfully extended this to include term based aspects as well. Hu et al. [32] proposed a hierarchical variant of xQUAD and PM-2, which models topics as a hierarchy instead of a list. Liang et al. [51] combined diversity with a data fusion task where a set of ranked lists are fused to generate a diversified ranked list based on topics generated using LDA. More recently, Sarwar et al. [84] proposed a linear programming formulation for topic proportionality used in combination with PM-2 for document diversification. Apart from the unsupervised models discussed above, some of the earlier work by Yue et al. [106] also included modeling explicit diversification as a supervised task using structured SVMs.

All the models described above are related to document diversification. In this thesis, we focus on answer passage diversification. A related work in this area is by Omari et al. [65], who demonstrated how xQUAD and PM-2 could be extended to generate diverse set of answers for CQA systems. However, they applied this to a setting where a small candidate set of answers were re-ranked using term clusters. In this dissertation, we explore this in an open-retrieval setting with clusters generated at passage level to perform answer passage diversification.

2.4 Conversational Search

One of the earliest work in Conversational Search was a system called I^3R proposed by Croft et al. [19], where users were provided with different search facilities, which could be selected based on the previous system output evaluations to satisfy their information needs. Later, Belkin et al. [5] introduced the model MERIT, which used scripted patterns of human-computer interaction to search for information.

Recently, Conversational Search has become very popular, with the advent of personal agents such as Siri, Cortana etc, where users could ask information seeking questions and carry on an interactive conversation with the agents. There is extensive work on Question Answering in a Machine Comprehension setting [36, 80, 79, 45], which could be considered as single-turn conversation, where annotators were provided with a question and passage and asked to indicate the answer span within it. This has been extended to a multi-turn setting with the introduction of datasets such as CoQA [82] and QuAC [11], to enable research in conversational question answering. A model using CoQA was introduced by Reddy et al. [82], where a combination of Pointer-Generator network (PGNet) [85] and Document Reader (DrQA) model [10] was used to identify answer spans and generate answers respectively. Huang et al. [33] combined the previous question/answers and their intermediate context representations to capture the conversational history and demonstrated its effectiveness using CoQA and QuAC datasets. Qu et al. [74] introduced a BERT based model, where a layer was added to indicate if the token at that location occurred in the conversational history. They also proposed an attention model [75] to capture the relative importance of various history contexts, instead of adding a fixed set of previous turns. Later, OR-QuAC dataset was introduced by Qu et al. [72] to adapt QuAC to an open-retrieval setting. They proposed an end-to-end model which retrieved top passages for each question and then outputted best answer spans. This was also extended to a weakly supervised setting [71].

Response ranking models have also been studied in conversational search, which focuses on re-ranking a small set of response candidates. Yang et al. [103] proposed a deep learning framework, which leveraged external information such as pseudo-relevance feedback and question/answer correspondence match for this task. They also proposed an intent-aware model [102], which weighted the utterances to better capture contextual history information. Another area of research has been to predict

clarification questions in response to user needs expressed in the form of short queries. Aliannejadi et al. [1] introduced the Qulac dataset for this task, which was developed to create clarification questions to cover various facets of the query. Hashemi et al. [31] introduced a model which used external sources such as top retrieved documents and a set of clarifying questions for the query. Bi et al. [6] proposed a MMR-BERT model to leverage negative feedback in conversations for improving the intent identification task. TREC Conversational Assistance Track (CAsT) [21, 20] is another effort in creating high quality test data and training resources to develop research in the area of open-retrieval conversational systems. In this dissertation, we show how clustering models could be used to improve conversational search task, by extracting useful response candidates and incorporating additional informational from them to update the queries.

CHAPTER 3

DATA COLLECTION

Current research in IR [101, 94, 13, 14, 15, 64, 66] focuses on passage relevance ranking, where higher ranked passages are considered answers to questions. To facilitate this research, various datasets [38, 13, 62, 16, 30] have been created for training and evaluation purposes, where the data annotation task consists of assigning relevance labels to passages with respect to the question. In this work, we focus on a different task, that of creating groups of answers, where the groups correspond to different answer types for a question. For example, for the question, "How can we get rid of skunks from the backyard?", the different answer types would be "Calling animal control" and "Using pest repellents" with answer passages assigned to one or both these groups. This task is fairly unexplored with only a single existing dataset, YahooL29 [65], which consists of questions from the Health Category of the Yahoo Answers forum. However, it suffers from a couple of limitations. The questions in this dataset cover only a single domain in the question-answering (QA) forum with the answer groups created using prepositional phrase clusters extracted from the answers without comparing full answer passages.

To mitigate these shortcomings, we create a new dataset which covers a more extensive set of domains from the question-answering (QA) forum. To this end, we conduct a manual annotation using a set of questions from the test collection of the publicly available ANTIQUE [30] dataset. For each question, the dataset creation process consists of two steps :

- Similarity label annotation between pairs of answer passages.

- Answer type generation using the similarity judgements constructed in the previous step.

3.1 Answer Passage Similarity Annotation

We first describe the creation of NFPassageQA_Sim[1] dataset, which contains similarity annotations between a pair of answer passages for a question.

3.1.1 Annotation Task Definition

To establish clear guidelines for annotation, we first describe the similarity task definition. In general, the definition of similarity is vague and could be interpreted in multiple ways. For example, the two passages "`Diabetes can be managed by good diet and exercise`" and "`Diabetes is a hereditary disease which affects about 5% of the population`" both refer to "`diabetes`" and can be interpreted as similar. However, they would be considered dissimilar given a context, such as a question "`How to treat diabetes?`", since the second passage doesn't answer the question. To eliminate such ambiguities and to ground the annotation process, we only include relevant passages for annotation. A formal definition of the task is given below.

Given a question q and a corresponding set of relevant answer passages $\mathcal{P}$, the data annotation task involves assigning a similarity label to passage pairs $(P_i, P_j) \in \mathcal{P} : P_i \neq P_j$. Here we assume the relationship to be symmetric i.e., $sim(P_i, P_j) = sim(P_j, P_i)$.

3.1.2 Data Annotation Process

As the input to the annotation process, we used the questions from the test collection of the publicly available ANTIQUE [30] dataset. A subset of 128 questions,

[1]https://ciir.cs.umass.edu/downloads/NFPassageQA/

which contains at least 10 relevant answers with labels {3,4} (similar) was filtered from the test collection. For each question q with n relevant answers, we create a set with m items $\mathcal{I} = \{(P_i^1, P_j^1), .., (P_i^m, P_j^m)\}$, where $m = \frac{n(n-1)}{2}$, consisting of all possible relevant answer pairs.

We employed workers from the Amazon Mechanical Turk (MTurk)[2] platform to perform the annotation. The workers were required to have a HIT (Human Intelligence Task) approval rate of 98% or higher, a minimum of 10000 approved HITs and be located in US, Canada, Australia or Great Britain, to narrow down native english speakers with a good performance record. They were paid $0.13 per HIT. Each input triple consisting of a question and a corresponding answer pair were assigned to three different workers. Detailed labeling instructions with examples were also provided to aid them with the task. After reading through the instructions, they assign a similarity label $0 - 4$ to the triple. The label definitions with examples are illustrated in Table 3.2. The data collection was performed in 7 batches. The label with a majority agreement among the workers was chosen as the ground truth. For cases with no majority agreement or with a majority label of 0, another round of annotation was performed to break the tie. We perform a set of filtering steps to remove instances that do not have sufficient agreement among the workers. First, we discard the instances where the ground truth couldn't be determined even after the second round of annotation. This also includes cases with a majority label of 0. Next, for those instances where we obtain a majority label, we remove cases where there isn't a majority agreement in terms of overall similarity. For example, an instance with votes [1,2,3,3] has a majority label agreement for label 3 but does not have agreement based on overall similarity (which is 2:2). The filtering brings down the

[2]https://www.mturk.com/

overall number of instances from 18629 to 18314. The final statistics of the dataset are shown in Table 3.1.

To ensure annotation quality, we added test triples with highly objective labels into each batch. This helps us identify workers who randomly click on labels without reading the instructions. We also conducted manual checks on 10% of the data to determine the quality of the annotation process. After identifying and rejecting around 8% of spurious data in the initial batches, we established a fully closed qualification restricted to around 70 workers in the subsequent batches. A Label 0 ("Not Sure") was also added to discourage workers from assigning a random label when unsure about the answer, especially due to a lack of domain knowledge.

Table 3.1 NFPassageQA_Sim dataset statistics.

#Questions	128
#Triples	18,314
#Label 4	245 (1.34%)
#Label 3	4,479 (24.45%)
#Label 2	12,724 (69.47%)
#Label 1	866 (4.72%)
AvgLen Questions	9.4
AvgLen Passages	61

3.1.3 Discussion

The descriptions of the labels with examples is illustrated in Table 3.2. Labels $3, 4$ indicate high similarity, indicating answers belonging to the same type, while label 2 indicates the passages belong to different answer types. Label 1 was added to capture any non-relevant passages, incorrectly labeled as relevant. Label 0 was added to reduce annotation noise and was removed from the final set of judgements.

Table 3.2 Label Descriptions

Label Type	Label	Description/Example
Similar	4	Both passages answer the question Both passages contain the same information, however they maybe worded differently Question: What do you mean by weed? Passage 1: Weed could mean the bad thing that grow in the garden or back and front yard or it could mean the drug Passage 2: It could mean weeds outside on the lawn or the drug
	3	Both passages answer the question. The passages belong to the same answer type. They may also contain information associated with a different type or other non-relevant information Question: What do you mean by weed? Passage 1: Weed could mean the bad thing that grow in the garden or back and front yard or it could mean the drug Passage 2: Marijuana and lots of it
Dissimilar	2	Both passages answer the question. The passages belong to different answer types Question: How can i get a cork out of,not into a wine bottle without a corkscrew? Passage 1: Use a screwdriver to put a wood screw into it, then pull the wood screw out with a pair of pliers, better yet get a $1 corkscrew Passage 2: If you have a syringe you can push it through the cork to the inside of the bottle press the air into the bottle and the pressure inside will force the cork out.
	1	At least one of the passages does not answer the question Question: How to cook Angus Burger? Passage 1: I usually cook burgers until they quit bleeding on both sides, then maybe just a little longer the cooking time will vary depending on the thickness of the burger. Passage 2: I'm not sure what the difference is other than the difference between a houstine and a angus but just because a bull is castrated doesnt make him an ox it just makes him a steer.
Not Sure	0	Not sure about the answer

Table 3.1 reports the final data statistics. Around 70% of the annotations correspond to Label 2, while Labels 3 and 4 cover around 26%. The high percentage of Label 2 is not surprising, considering the nature of the data used for annotation. The questions were extracted from a CQA discussion forum, where users tend to give alternate answers to the questions. Predictably, Label 4 occurs very infrequently in

the dataset (since, most answers would are unlikely to paraphrases of each other). We also performed manual checks to confirm that the answers were not being mislabeled as non-relevant (Label 1), since it has a relatively high coverage (5%).

3.2 Answer Type Generation

In this section, we describe the process of generating NFPassageQA_Div[3] dataset, which contains answer passage groups where each group corresponds to an answer type.

3.2.1 Formal Task Definition

The passage similarity annotations provide us with similarity values pertaining to all pairs of relevant passages corresponding to each question. This information can then be used to generate answer types (or subtopics) for these questions. A formal definition is given below:

Given a question q, a corresponding set of relevant answer passages $\mathcal{P}$ and a set of similarity annotations between passage pairs $sim(P_i, P_j)$, where $P_i \in \mathcal{P}$ and $P_j \in \mathcal{P}$, the dataset creation task involves automatically identifying the various answer types (or subtopics) $\mathcal{T}$ and assigning passages in $\mathcal{P}$ to them.

3.2.2 Dataset Construction:

Since the similarity annotations contain a relatively high ratio (5%) of non-relevant answer passage pairs, the first step is the identification of non-relevant passages, which must be removed to reduce noise. To identify them, all passages appearing in more than 40% of passage pairs, corresponding to each question and annotated with Label 1 is marked as non-relevant, which resulted in the removal of 596 passage pairs. This process was also manually cross-checked to ensure that no relevant passages were

[3]https://ciir.cs.umass.edu/downloads/NFPassageQA/

discarded. Due to the change in number of relevant passages per question, we retain only questions with at least 10 relevant passages remaining after the previous step, which reduces the number of questions to 93. Based on the similarity labels with values 3 and 4, we next construct all possible passage combinations and identify the longest non-overlapping passage clusters for each question. These are considered to be the answer types (or subtopics). Each of the remaining passages are then added to the answer type if at least one of the passages in the cluster is similar to it.

Relevance judgements are then assigned to each passage with respect to each answer type. The original passages within each non-overlapping cluster (representing a unique answer type) and other passages subsequently added based on partial similarity to original cluster elements are assigned a relevance judgement of 1 indicating `relevance`. All the other passages are considered `non-relevant` with value 0.

Table 3.3 Example Answer Types

Question	How do I get rid of mice humanely?
Answer Type 1 *(Use traps)*	Home Depot sells live traps.... Put down a humane trap.... Get a cat or use mouse traps...
Answer Type 2 *(Use natural predators such as cats)*	Just get or borrow a friend's cat... invite my cat over,she is a great mouser! Get a cat or use mouse traps...

Table 3.4 NFPassageQA_Div Answer Type Distribution

#Types	1	2	3	4	5
#Questions	39 (41.9%)	32 (34.4%)	15 (16.1%)	4 (0.04%)	3 (0.03%)

3.2.3 Discussion

Table 3.3 gives an example of a question and with two corresponding answer types. A shortened version of the passages assigned to the answer type is given next to it.

Due to the nature of the dataset, a passage can belong to multiple answer types. For instance, the last passage in both answer types in Table 3.3 corresponds to both `Traps` and `Cats`. Table 3.4 gives the distribution of answer types corresponding to the questions. As indicated in the table, about 58% of the questions have multiple answers associated with it. We also performed manual checks to check the overall effectiveness of the data generation process.

3.3 Summary

In this chapter, we describe the process of creating evaluation datasets for answer passage similarity (NFPassageQA_Sim) and answer type generation (NFPassageQA_Div) tasks. Here, the answer passage similarity task is defined in terms of whether a pair of passages belong to the same answer type or not. We first describe the manual annotation process used to collect the answer passage similarity annotations for 128 questions from the Yahoo CQA forum. Next, we cover the details about the labels used and the various data processing steps performed to complete this annotation. We then describe how these similarity annotations were used to assign answer types to passages for a subset of 93 questions. We also provide statistics and a brief discussion about the data. In summation, the process described in this chapter demonstrates how annotation of QA datasets with answer type information can be achieved indirectly by a two-step process.

CHAPTER 4

ANSWER PASSAGE SIMILARITY CLUSTERING

In the previous chapter, we discussed the annotation process for collecting similarity-based labels for pairs of answer passages given a question, and created an evaluation set for the answer passage similarity task. The next step is to use this dataset to evaluate and determine an effective strategy to create answer similarity clusters.

We adopt kNN (k nearest neighbor) used in cluster based settings [57, 96, 58, 44, 43, 78] to find the most similar passages corresponding to an answer passage. However, in these traditional settings, only term-based representations such as Language Models (LM), were used for creating the clusters. Recent work using distributed representations such as Glove [68] and Word2Vec provides a different type of representation which could be used for clustering, one that would incorporate term context in addition to term count information. Besides these, longer contextual information, which looks at more than word contexts such as sentence or passage level contexts in models such as BERT [25], could also be useful in determining passage similarity. Hence, we explore GloVe and BERT models along with LM to generate passage level representations.

Besides the unsupervised models, we also study how weak supervision can be used to generate representations for this task. Older weak supervision models [24, 108] typically employed huge amount of weak data for training in order to attain significant improvements. However, using a large pre-trained model such as BERT, which implicitly captures basic semantic and syntactic properties of text is highly useful in providing a good starting point to build more task-specific models, using

less training data. Therefore we use BERT for training and study the impact of using different types of pseudo-labels and its impact on this task.

4.1 Task Definition

Figure 4.1 illustrates the high-level architecture of the task. This is similar to the setting used in cluster based models [57, 96, 58, 44, 43, 78]. Given a question q and a passage collection C, a standard retrieval model can be used to retrieve a list of top n passages $\mathcal{P}$. A cluster is generated for each of these passages $P_i \in \mathcal{P}$ with respect to every other passage $P_j : P_i \neq P_j$. An effective clustering method must be able to perform the following:

- Cluster relevant answer passages together
- Cluster relevant answer passages of the same type together

To generate the clusters, we determine the nearest neighbors corresponding to each passage and rank them based on the similarity score.

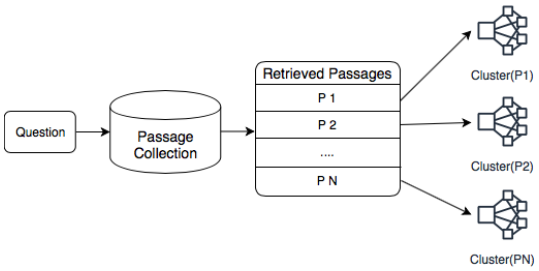


Figure 4.1 Answer Passage Clustering pipeline

4.2 Answer Passage Clustering

Different types of representations and comparison strategies result in different types of clusters. A comparative study would help to determine which is most effective for this task. Therefore, we study the impact using three different unsupervised text representations and BERT-based weakly supervised representations.

4.2.1 Unsupervised Representations

4.2.1.1 Types

4.2.1.1.1 Language Model with Dirichlet smoothing (LM): Language Model is selected as the term frequency based representation. An alternative representation is the tf-idf, however we choose LM since it has been demonstrated to perform better in cluster based settings [44]. Given a passage P, the language model can be estimated using maximum likelihood estimation: $Pr_P^{MLE} = \frac{c(w,P)}{|P|}$ where $c(w, P)$ is the count of word w in passage P. Dirichlet smoothing [109] can be applied to interpolate this estimate with collection (C), $Pr_P^{Dir(\mu)} = \frac{c(w,P)+\mu Pr(w|C)}{|P|+\mu}$. These values are calculated at passage (P) level. Since questions can provide a context to clustering, we also experiment with question augmented versions ($P + q$) to determine if question terms can improve the clustering performance.

4.2.1.1.2 Global Vectors for Word Representation (GloVe): Instead of using term frequency statistics to represent text, models such as GloVe [68] incorporate global term co-occurrence counts along with the local contextual information to create distributed representations. To generate passage representations, the vectors are combined via averaging ($E[P]$) or by idf-weighting ($\sum idf*P$). The sentence representations are created by averaging the word vectors. The sentences within a passage are then averaged to generate another set of passage representations ($E[s \in P]$). The question augmented versions are also generated by including question term vectors ($E[P + q]$, $\sum idf * (P + q)$, $E[s \in (P + q)]$).

4.2.1.1.3 Bidirectional Transformers for Language Understanding (BERT): Instead of focusing on term statistics or local word contexts, longer sequence context information can be used to model representations at sentence/passage level. BERT [25] representations are conditioned on both left and right contexts across all

the layers of a deep neural model. For each token, BERT generates an embedding using position, segment and token embeddings. BERT pre-training is performed using two unsupervised tasks: Masked Language Modeling and Next Sentence Prediction. The passage/sentence representation is generated by giving a single input (P) and the question augmented versions are generated by giving the question terms as the second input ($P + q$). A second set of passage representations are also generated by averaging the sentence representations ($E[s \in P]$, $E[s \in (P + q)]$). We experiment with two different variants of the BERT output: $[CLS]$ and $[TOK]$. The first case corresponds to the $[CLS]$ token embedding outputs (P $[CLS]$, $P+q$ $[CLS]$, $E[s \in P]$ $[CLS]$, $E[s \in (P+q)]$ $[CLS]$) and the second corresponds to combining individual token representations via averaging ($E[P]$ $[TOK]$, $E[P+q]$ $[TOK]$) or by idf-weighting ($\sum idf * P$ $[TOK]$, $\sum idf * P + q$ $[TOK]$).

4.2.1.2 Answer Comparison

Besides text representations, different metrics for comparison also generate different sets of nearest neighbors. To calculate the similarity score, $sim(P_i, P_j)$ between two passages P_i and P_j, LM models [78, 87] employ the cross entropy ($\mathcal{H}$) similarity between the Maximum Likelihood ($Pr_{P_i}^{MLE}$) and Dirichlet ($Pr_{P_j}^{Dir(\mu)}$) estimates: $sim(P_i, P_j) = \exp(-\mathcal{H}(Pr_{P_i}^{MLE}, Pr_{P_j}^{Dir(\mu)}))$. For both GloVe and BERT models, Euclidean distance is used as the score. For the question augmented versions, comparison is done by adding question terms to the passages.

Inspired by the success of sentence level models for passage retrieval [101], we also investigate a greedy strategy to set the passage level scores from the sentence level scores. Given passages P_i and P_j and sentences s_m and s_n : $s_m \in P_i$ and $s_n \in P_j$, $sim(P_i, P_j) = \max_{s_m \in P_i, s_n \in P_j} sim(s_m, s_n)$. In case of GloVe and BERT models, the minimum value of scores is used. As in earlier cases, we experiment with both sentence level ($Greedy_s$) and question augmented versions ($Greedy_{s+q}$).

4.2.1.3 Model Combinations

Apart from using scores from individual text representations to rank the passages, we also test to see if a combination of the scores from the various inputs can improve the performance. In order to aggregate the scores, a weighted summation of the individual scores is used. All three scores are first set to be in range $[0 - 1]$. To combine a pair of scores (LM+GloVe, LM+BERT, GloVe+BERT), the parameter λ is used for weighting: $Score_{aggr} = \lambda Score_1 + (1 - \lambda)Score_2 : 0 \leq \lambda \leq 1$. In order to aggregate three scores (LM+GloVe+BERT), parameters β and γ are used as follows: $Score_{aggr} = \beta Score_1 + \gamma Score_2 + (1 - \beta - \gamma)Score_3 : 0 \leq \beta \leq 1, 0 \leq \gamma \leq 1$.

4.2.2 Weak Supervision

Good training data is essential for supervised machine learning models to work effectively. Obtaining such high quality data involves significant annotation effort and incurs a high monetary cost. This is especially difficult for deep learning models, which give state of the art results for many tasks, but require a massive amount of labeled data for training. Recent work has demonstrated the use of weakly labeled data for improving core IR tasks such as document ranking [24]. However, such models require millions of weak labels to learn the ranking function. Xu et al. [100] showed that large pre-trained models such as BERT [25], fine-tuned with small number of weak labels aggregated from different sources can be used for passage ranking. The fine-tuning process forces the model to learn task specific relationships, even from weakly labeled data, which explains its effectiveness.

We adopt the weak supervision strategy and generate pseudo-labels, to learn an improved similarity function and semantic vector representation for passage clustering, using the BERT [25] model. We use pseudo-labels from three main sources : term based Language Models (LM) and semantic models GloVe and pre-trained BERT, and show how these can be used to learn better similarity functions for this task. The

following sub-sections describe the process of generating the pseudo-labels and the model architecture.

4.2.2.1 Pseudo-Labelers

Formally, the pseudo-labeling process can be defined as follows. Start with a question q and create a ranked list of answer passages $\mathcal{P}$. Then, for a passage $P_i \in \mathcal{P}$ with $rank_{QL}(q, P_i) \leq 10$, the weak labeling process generates a list of n passages, ranked based on the similarity score, $\mathcal{R} = (P_j^1, P_j^2, ...P_j^n)$ with $P_j^k \in \mathcal{P}$ and $\forall P_j^k, P_j^k \neq P_i$. For our experiments, the list $\mathcal{P}$ is created using the Query Likelihood (QL) model and the top $n = 200$ passages are clustered, both reflecting the typical setting for cluster based models [78, 87] and so the convention we use for our experiments. We experiment with the three weak labeling functions based on the different text representations explained in the earlier sections.

4.2.2.1.1 Language Model with Dirichlet smoothing (LM): We represent a passage using the Document Language Model with Dirichlet smoothing as described under Section 4.2.1.1.1 to create representations which capture term frequency based information. The similarity score between a pair of passages is calculated using cross-entropy as given in Section 4.2.1.2.

4.2.2.1.2 Global Vectors for Word Representation (GloVe): As described in Section 4.2.1.1.2, we use GloVe [68] representations to incorporate global term co-occurrence counts along with the local contextual information to create representations which capture semantic relationships along with term statistics. We generate passage representations by combining the vectors using idf-weighting of terms present in the question as well as the passage. Question terms are also used to add contextual information necessary for clustering and was found to perform better. We use Euclidean distance as the scoring function.

4.2.2.1.3 Bidirectional Transformers for Language Understanding (P-BERT):

As described in Section 4.2.1.1.3, BERT representations are used to capture contexts longer than word or phrase-level for the clustering task. The vector representations corresponding to a passage is generated by giving two inputs : question q and passage P. Similar to generating GloVe representations, we use question terms in addition to passage information to provide more context. The [CLS] token embedding representation corresponding to the input sequence is used for this task. The scoring function is the Euclidean distance.

4.2.2.1.4 Combinations: We also experiment with different combinations of the three previously defined pseudo-labelers to generate four new weak signals given in Section 4.2.1.3 : LM+GloVe, LM+P-BERT, GloVe+P-BERT and LM+GloVe+P-BERT. We also experimented with a majority voting technique for combining labels from different sources to create high quality pseudo-labels for training. However, the number of training instances for which there was agreement between even two of the sources was too low for training.

4.2.2.2 Model Architecture

The BERT [25] model is used as the framework for the weak supervision experiments. For an input question q and passage pair (P_i,P_j), the model must learn a similarity function and output a score for the triple. The training for this task is performed by feeding the inputs to BERT and fine-tuning the model based on two different loss functions.

4.2.2.2.1 Point-wise model: The point-wise loss function is the default cross entropy loss used for the sentence pair classification experiments in BERT [25]. The architecture is shown in Figure 4.2. The two inputs to the model are $(q + P_i)$ and $(q + P_j)$ where "+" indicates that question terms have been concatenated with the

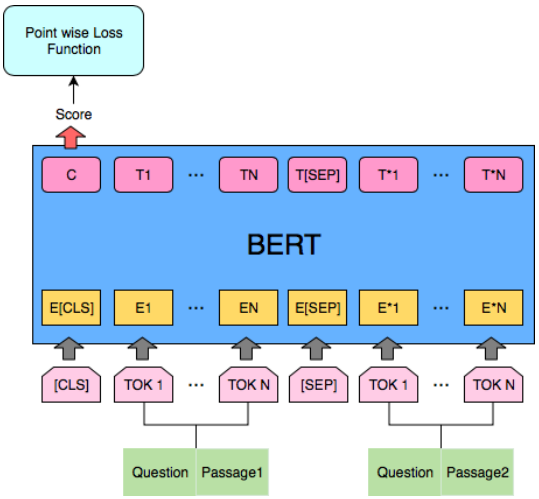


Figure 4.2 Point-wise model trained with default cross-entropy loss function

corresponding passage terms. For a triple (q, P_i, P_j) with $\hat{s}(q, P_i, P_j; \theta)$ as the scoring function learned by the model under the parameters θ and $s(q, P_i, P_j)$, the ground-truth generated by the weak labeler, the training loss can be defined as follows :

$$\mathcal{L}(q, P_i, P_j; \theta) = s(q, P_i, P_j) \log \hat{s}(q, P_i, P_j; \theta) \tag{4.1}$$

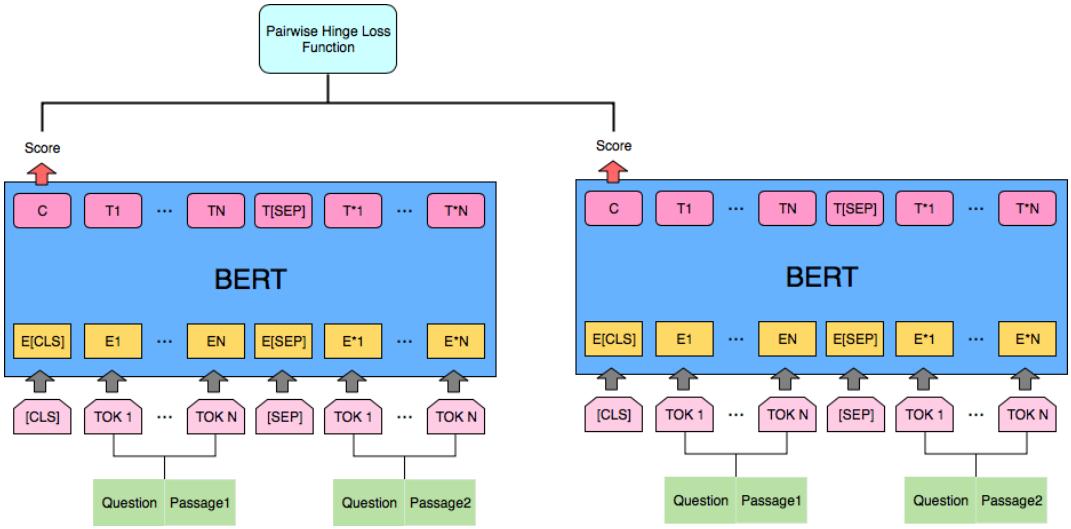


Figure 4.3 Pair-wise model trained using hinge loss function

4.2.2.2.2 Pair-wise model: The pair-wise model is similar to the Rank model [24] and the passage ranking model described by Xu et al. [100]. The loss function employed is the pair-wise hinge loss function. The architecture is shown in Figure 4.3. For an input pair, $[(q, P_i, P_j), (q, P_i, P_j^{'})]$ with point-wise scoring functions $\hat{s}(q, P_i, P_j; \theta)$ and $\hat{s}(q, P_i, P_j^{'}; \theta)$, and the weak labels $s(q, P_i, P_j)$, and $s(q, P_i, P_j^{'})$, the model is trained to minimize the hinge loss as follows:

$$\mathcal{L}(q, P_i, P_j, P_j^{'}; \theta) = \max\{0, \epsilon - sign(s(q, P_i, P_j) - s(q, P_i, P_j^{'})) \\ (\hat{s}(q, P_i, P_j; \theta) - \hat{s}(q, P_i, P_j^{'}; \theta))\} \tag{4.2}$$

The point-wise model[1] used in this case is different from the default BERT model. The inputs to the model are same as the default version, $(q + P_i)$ and $(q + P_j)$ where "+" indicates that question terms have been concatenated with the corresponding passage terms. The BERT scoring model generates hidden states for the [CLS] token for the input and the final hidden layer is fed into a dense layer. We consider two variants of the model, one with a linear output activation (Pair-wise Linear) and the other with tanh activation (Pair-wise tanh). During test time, the corresponding point-wise scores are used to generate the similarity scores.

Table 4.1 Data Statistics

Dataset	#Question	#Pass	Avg Rel Pass/Question	Avg Question Len	Avg Rel Pass Len
NFPassageQA_Sim	128	2098	16.5	9.4	61

[1]References to point-wise model throughout the rest of the chapter indicate the default BERT model

4.3 Experimental Setup

4.3.1 Data Overview:

The evaluation of the answer passage similarity experiments are conducted using the NFPassageQA_Sim dataset. The data statistics are presented in Table 4.1. For weak supervision experiments, we used the ANTIQUE dataset collection for our experiments. This collection consists of short passages which are answers to non-factoid questions, introduced by Cohen et al. [13], called nfL6[2]. We randomly sampled 200 questions from the training set of the dataset to create a validation set. The remaining 2226 questions were used for training. We use the questions from the newly created NFPassageQA_Sim dataset to evaluate the models based on how well relevant passages of the same type cluster together. To evaluate the performance of relevant passage clustering, the relevance judgements for the corresponding questions from the ANTIQUE [30] test set is used. The process to generate weak labels for the point-wise and the pair-wise models is discussed below.

Training and Test Setup: For training, the pseudo-labeling process described

Table 4.2 Weak Supervision Experimental settings

#Train questions	# Train point-wise instances	# Train pair-wise instances	#Test questions
2226	222600	400000	128

in Section 4.2.2 is used to generate a ranked list of passages for each (q, P_i) pair. Instead of adding the set of all P_j to training data, we add only the top 10 passages as positive/negative examples. To create weak labels for point-wise models, the top 5 passages from the ranked list are labeled as positive (1) and the next 5 passages are labeled as negative (0). For the pair-wise models, we need a pair of passages from the ranked list to create the training instances. These pairs are generated using

[2]https://ciir.cs.umass.edu/downloads/nfL6/

a sliding window method. For each passage in the ranked list, the next 5 passages below it in the ranked list are considered to have lower scores and added as training data. From these generated instances, we randomly sample a subset for our experiments. At test time, we follow the same initial process described in Section 4.2.2. The point-wise scores are generated for each test triple (q,P_i,P_j) and the clustering is performed based on these scores. We perform clustering over all passages P_j – i.e., a set of 200 passages – for each P_i with $rank(P_i) \leq 10$ (same as the settings in Section 4.2.2). The baseline methods also follow the same convention for correct comparison. The experimental settings are summarized in Table 4.2.

4.3.2 Implementation Details

For the unsupervised representations, the initial processing of passages and questions for creating the Language Model (LM) model are the same as [87]. The passages and questions are stemmed by a Krovetz stemmer [41]. The stopwords on the INQUERY list [2] are removed from the questions. 3-fold cross-validation was performed to set μ, λ, β and γ parameters for test questions. For the GloVe experiments, we used 300d pre-trained GloVe [68] vectors[3] and for the BERT based experiments [25], the final layer hidden vectors generated using BERT-Base (Uncased) pre-trained model[4] were used. In order to combine scores from various models, the output scores from the best performing model from each type were used.

In case of weak supervision experiments, the following default parameter values were set for the train questions during the pseudo-labeling process : LM: $\mu = 10$, LM+GloVe: $\lambda = 0.3$, LM+P-BERT: $\lambda = 0.3$, GloVe+P-BERT: $\lambda = 0.2$, LM+GloVe+P-BERT: $\beta = 0.1$ and $\gamma = 0.1$ with the processing steps as above. GloVe and BERT vectors are generated the same as given above. For BERT, we also experimented

[3]https://nlp.stanford.edu/projects/glove/

[4]https://github.com/google-research/bert

with other layers (other than Layer 1) as well as various ways of combining, such as averaging and concatenating and found the Layer1 vectors to be most effective for this task. The models trained on BERT are implemented using TensorFlow[5] and fine-tuned after initializing with the BERT-Base (Uncased) pre-trained model. The maximum sequence length is set to 128, with each input truncated to length 64. The batch size is set to 20. The initial learning rate was selected from $[1e^{-5}, 2e^{-5}, 3e^{-5}]$ by tuning on the validation set. The dropout parameter is set to 0.1. The experiments were conducted on a single GeForce GTX 1080 GPU. The train time for the point-wise model was around 1 hour and pair-wise models took around 2 hours for training. The inference time was around 50 minutes for both model types.

The k-nearest neighbor (kNN) clustering was conducted using the kDTree algorithm in the sklearn toolkit. The typical setting for cluster based models [78, 87] is to cluster only the top n documents/passages retrieved for a question. We follow the same procedure, where n is set to 200.

4.3.3 Evaluation

To effectively evaluate the model, we need to assess how well the passages cluster together for the various models. Besides this, ranking metrics (instead of typical clustering metrics) need to be used to determine if the models return relevant passages in higher ranks. We use Precision@k, NDCG@k, MRR, Recall@k for evaluation where k=10,20. For each question, for every relevant passage, we evaluate based on the metrics and average the values across all relevant passages. The values are averaged across the questions to get the final value. The same process is followed for non-relevant passages.

[5]https://www.tensorflow.org/

Table 4.3 Results on NFPassageQA_Sim dataset for clustering relevant passages of the same type.*,◇ and † indicates significance with respect to strongest LM, GloVe and BERT models respectively.

Model	Level of Comparison	Method	NFPassageQA_Sim		
			P@10	NDCG@10	MRR
LM	Passage	P	0.0523	0.0853	0.1516
		$P+q$	0.0430	0.0687	0.1356
	Sentence	Greedy_s	**0.0633**	**0.1073**	**0.1775**
		Greedy_{s+q}	0.0477	0.0844	0.1691
GloVe	Passage	$E[P]$	0.0670	0.1171	0.2039
		$E[P+q]$	0.0537	0.1035	0.1696
		$\sum idf * P$	**0.0779***	**0.1401***	**0.2407***
		$\sum idf * (P+q)$	0.0663	0.1244*	0.209
	Sentence	Greedy_s	0.0536	0.0941	0.1665
		Greedy_{s+q}	0.0360	0.0666	0.1212
BERT	Passage	P $[CLS]$	0.0544	0.1170	0.1825
		$P+q$ $[CLS]$	**0.0838***	**0.1925**$^{*\diamond}$	**0.2648***
		$E[P]$ $[TOK]$	0.0665	0.1251*	0.2065
		$E[P+q]$ $[TOK]$	0.0754*	0.1461*	0.2311*
		$\sum idf * P$ $[TOK]$	0.0747*	0.1434*	0.2425*
		$\sum idf * (P+q)$ $[TOK]$	0.0820*	0.1568$^{*\diamond}$	0.2386*
	Sentence	$\text{Greedy}_s[CLS]$	0.0359	0.0699	0.1242
		$\text{Greedy}_{s+q}[CLS]$	0.0360	0.0755	0.1384
COMB	Combination	LM+GloVe	0.0791*	0.1361*	0.2250*
		LM+BERT	0.0856*	0.1895$^{*\diamond}$	0.2572*
		GloVe+BERT	**0.0952**$^{*\diamond\dagger}$	**0.2046**$^{*\diamond\dagger}$	**0.2936**$^{*\diamond\dagger}$
		LM+GloVe+BERT	0.0836*	0.1532$^{*\diamond}$	0.2561*

4.4 Results and Analysis

In this section, we analyse the output results generated from unsupervised representations and weak supervision method.

4.4.1 Unsupervised Representations

Table 4.3 reports results for clustering relevant passages of the same type for the NFPassageQA_Sim dataset using various unsupervised representations described in Section 4.2.1. The results corresponding to averaged sentence representations ($E[s \in P]$, $E[s \in P+q]$) have not been added to the table since they do not perform as

well as the passage based models ($E[P]$, $E[P + q]$). The results demonstrate that the BERT pre-trained model with question and passage as inputs improves over both GloVe and LM models in all cases. Between the two variants of the BERT model, BERT $[CLS]$ token based representation performs the best. The idf weighted BERT $[TOK]$ representation also performs comparably in most cases. Though the greedy method works the best for the LM model, it performs poorly in case of both GloVe and BERT models, which indicates the usefulness of passage-level contextual information in determining similarity. Question augmented models are generally more effective than models with only passage level information for both GloVe and BERT. Though NFPassageQA_Sim metric values show the best performance for passage based model for GloVe, it is not significantly better than the question augmented model. We conjecture that the slight improvement is due to the setup used for this dataset, where the top 200 passages retrieved using the Query Likelihood baseline are used for clustering and have many terms in common with the question. However, for this case, the question-augmented BERT model significantly improves clustering over the passage based model indicating that the BERT model captures more than term level contextual information. In this work, we have considered a LM baseline which is in line with what was used in Cluster based IR retrieval models which aligns with the similarity task. However, other LM models could display different behaviours and capture more information. This is left as future work.

4.4.1.1 Qualitative Analysis

We also conduct a qualitative analysis of the output from the various models. Table 4.4 illustrates an example, where the top ranked passage from the best performing BERT, GloVe and LM models are shown. The LM model retrieves an unrelated passage on the basis of the high text overlap of the terms "`to get rid of`" with the input passage. Since the best performing LM models are not question-biased, term

matching with respect to the question is not performed. The GloVe model identifies a related passage with some term overlap of terms such as "`wash your face, oil, face, pimples`" and other related words such as "`acne`". As expected, the BERT model identifies a related passage with minimum term matching but with a high semantic match across the entire passage. This shows that the LM similarity model retrieve passages without considering any contextual information, while GloVe models consider more context by looking at related words and the BERT model is able to capture wider contexts at sentence or paragraph level and therefore able to retrieve more similar passages.

Table 4.4 Top 1 nearest neighbors for the input question and passage across different models

Question	How can I **get rid of pimples** on my back?
Passage	There are lotions you can get **to get rid of pimples**. You can **wash your face** with soap to get rid of pre-pimples. **Pimples** are caused by **oil** being built up on the skin.
BERT	There are several at home remedies to **get rid of pimples**. Toothpaste, the fluoride in it dries out the puss thus shrinking the pimple. Raw piece of potato over pimple. Some ingredient in a potato sucks puss and dirt out of the skin. I've been told that you can feel it working. If you want to try something over the counter then try proactiv.
GloVe	Drink a lot of water. Atleast 8 glasses a day. **Wash your face** regularly with an **oil** free **face** wash and avoid eating oily food. All these put together will help you get rid of **pimples** and acne.
LM	First of all a salamander is family of the lizard and they are not dangerous and they are totally harmless, and why on earth would you want **to get rid of** them they eat insects, bugs and all types of insects, by the way there is no way that you are going **to get rid of** them, **get rid of** all the insects and maybe you will **get rid of** them.

4.4.1.2 Model Combinations

We also study the impact of linearly combining the various models together. The results in Table 6.3 indicate that GloVe and BERT combination significantly outperforms the best performing individual models. Based on the way it is combined, if a passage has a high similarity score with respect to two types of representations,

then it would get an overall higher score than passages which scores well with only one type. To test this further, we calculate the overlap between the top (10) nearest neighbors between each pair (LM, GloVe), (LM, BERT) and (GloVe, BERT) from the best performing models in the NFPassageQA_Sim dataset. For the first two cases, the overlap is around 6-7%, while the overlap is almost double (12%) between GloVe and BERT. This indicates that semantic models capture similar information and can be combined effectively creating an additive effect. However a model such as LM which captures term matching does not combine well with these models. Other alternate techniques may have to be investigated to study how this could be better integrated to get the best of both type of models.

4.4.1.3 Performance based on relevance metrics

Table 4.5 shows the performance of the various models while clustering relevant passages on the NFPassageQA_Sim dataset. We also performed the same experiments for non-relevant passages i.e, whether most similar passages are relevant or non-relevant. We found that non-relevant passages cluster well with other non-relevant passages and not with relevant passages. The table demonstrates that the results follow the same pattern as clustering passages of the same type.

Table 4.5 Results on NFPassageQA_Sim dataset for clustering relevant answers. *, $\diamond$ and † indicate significance with respect to strongest LM, GloVe and BERT models respectively.

Model	Level of Comparison	Method	P@10
LM	Sentence	Greedy_s	0.1330
GloVe	Passage	$\sum idf * P$	0.1607^{*}
		$\sum idf * (P + q)$	0.1478
BERT	Passage	$P + q$ $[CLS]$	$0.2279^{*\diamond}$
COMB	Comb	**GloVe+BERT**	$\mathbf{0.2361}^{*\diamond}$

4.4.2 Weak Supervision

Table 4.6 and Table 4.7 report the results corresponding to the three main sources of pseudo-labels: GloVe, LM and pre-trained BERT (described in Section 4.2.2) on the questions from the NFPassageQA_Sim dataset. Table 4.6 gives the performance of the models for the task of clustering relevant passages. Table 4.7 provides the performance of the models for clustering relevant passages of the same type. Pseudo-labels derived using GloVe perform the best, with all three models significantly improving over the corresponding baseline. Even though the models trained on pseudo-labels generated using Language Modeling information and pre-trained BERT perform worse than their corresponding baselines, the best performing GloVe model significantly outperforms both LM and BERT baselines, which shows the overall effectiveness of this method. In the following sections, we analyse the model performance in detail and study its various strengths and weaknesses.

Table 4.6 Weak Supervision results on NFPassageQA_Sim datasets for clustering relevant passages (Rel Clusters) for the three main pseudo-labelers. † indicates significance with respect to corresponding baselines. P-BERT refers to pre-trained BERT. The scores for the best performing trained model with respect to each baseline has been marked in bold.

Pseudo-labeler	Model	Rel Clusters			
		P@10	P@20	R@10	R@20
LM	Baseline	0.1057	0.0917	0.0960	0.1780
	Point-wise	0.0706	0.0624	0.0770	0.1478
	Pair-wise Linear	0.0803	0.0739	0.0813	0.1575
	Pair-wise tanh	**0.0871**	**0.0794**	**0.0937**	**0.1811**
GloVe	Baseline	0.1478	0.1114	0.1558	0.2468
	Point-wise	**0.2087**†	**0.1663**†	**0.2403**†	**0.4070**†
	Pair-wise Linear	0.2002†	0.1632†	0.2192†	0.3722†
	Pair-wise tanh	0.1982†	0.1631†	0.2204†	0.3895†
P-BERT	Baseline	0.2279	0.1745	0.2720	0.4580
	Point-wise	0.1874	0.1457	**0.1983**	0.3277
	Pair-wise Linear	0.1928	0.1479	0.1952	**0.3284**
	Pair-wise tanh	**0.1932**	**0.1498**	0.1868	0.3224

Table 4.7 Weak Supervision results on NFPassageQA_Sim datasets for clustering relevant passages of the same type (Sim Clusters) for the three main pseudo-labelers. † indicates significance with respect to corresponding baselines. P-BERT refers to pre-trained BERT. The scores for the best performing trained model with respect to each baseline has been marked in bold.

Pseudo-labeler	Model	Sim Clusters			
		P@10	P@20	R@10	R@20
LM	Baseline	0.0523	0.0418	0.0942	0.1537
	Point-wise	0.0380	0.0287	0.0660	0.1041
	Pair-wise Linear	**0.0427**	**0.0346**	**0.0779**	0.1225
	Pair-wise tanh	0.0420	0.0335	0.0738	**0.1321**
GloVe	Baseline	0.0663	0.0460	0.1384	0.1813
	Point-wise	**0.0964**†	**0.0686**†	**0.1887**†	**0.2736**†
	Pair-wise Linear	0.0924†	0.0674†	0.1799†	0.2617†
	Pair-wise tanh	0.0935†	0.0667†	0.1762†	0.2506†
P-BERT	Baseline	0.0838	0.0626	0.1662	0.2483
	Point-wise	0.0815	0.0549	0.1585	0.2118
	Pair-wise Linear	**0.0855**	**0.0594**	**0.1661**	**0.2399**
	Pair-wise tanh	0.0795	0.0578	0.1595	0.2215

4.4.2.1 Impact of pseudo-labeling source

The results indicate a wide disparity in performance, between the models trained with pseudo-labels obtained from term based semantic models such as GloVe and those trained with term count information such as LM. The pseudo-labels generated by the pre-trained BERT model were also not helpful in improving the performance. To reduce over-fitting on training data, we also implemented early stopping. The quality of pseudo-labels was also investigated to determine if this could account for the variability in performance across different types of signals. Since the training data was obtained from the ANTIQUE training set, we could evaluate if the relevant passages cluster well together based on the question, passage relevance judgements. Table 4.8 contains the performance of the weak labels generated for training the point-wise model. The results indicate that, for this setting, pseudo-label quality cannot be used as an indicator to determine the overall performance of the trained

model. Even though BERT label quality is better than GloVe based labels, the model trained with this data did not improve over the baseline. This indicates that term based semantic models such as GloVe provide information which combines well with BERT while fine-tuning and provides additional information which would be useful to improve the model.

Table 4.8 Pseudo-label Quality

Pseudo-label Source	P@5	P@10
LM	0.0651	0.0535
GloVe	0.0921	0.0661
P-BERT	**0.1797**	**0.1264**

4.4.2.2 Comparison between GloVe trained model and BERT/LM baselines

Results in Table 4.7 indicate the effectiveness of GloVe based inputs for fine-tuning BERT with respect to the corresponding baseline. We also compare the GloVe trained model with BERT and LM baseline. Table 4.9 indicates that the GloVe trained model performs significantly better than both LM and BERT baselines, which indicates the effectiveness of using GloVe signals in conjunction with BERT.

Table 4.9 Comparison of performance between Glove trained model and BERT baseline. † indicates significance with respect to the BERT baseline and ⋄ indicates significance with respect to the LM baseline. The performance is based on the clustering performance of the relevant passages of the same type.

Models	P@10	P@20
LM baseline	0.0523	0.0418
BERT baseline	0.0838	0.0626
GloVe point-wise model	**0.0964**$^{\diamond}$†	**0.0686**$^{\diamond}$†

4.4.2.3 Comparison between different model types

Based on the results in Table 4.6 and Table 4.7, the different model types : point-wise and pairwise behave comparably. The Win/Loss statistics based on Precision@20 measuring the clustering performance of relevant passages of the same type, against the best baseline/model pair is shown in Table 4.10. In general the linear models perform better than the non-linear variant.

Table 4.10 Win-Loss statistics for GloVe models compared with the baseline for clustering relevant passages of the same type, with respect to P@20

Models	W/L Questions	W/L Rel Passages
Point-wise	**56/12**	**168/77**
Pair-wise Linear	51/13	165/77
Pair-wise tanh	53/15	160/77

4.4.2.4 Contribution of question/passage relevance

We also conducted an experiment to estimate how well the models trained with (question/ passage) relevance judgements perform compared to these models. Relevance judgements provide a very coarse grained definition of similarity. For instance, as per the label definitions in Table 3.2, a relevant and non-relevant passage can be considered dissimilar. But it is unclear how well other gradations of similarity information is captured. We first train a BERT model on the relevance judgements from the training set in the ANTIQUE dataset, leaving out the validation questions. The relevant labels 3,4 are mapped to 1 and labels 1,2 are mapped to 0. The resulting trained checkpoint file is then used to infer vectors corresponding to the test set questions and passages. Using these vectors, we generate nearest neighbors and perform the evaluation.

The results reported in Table 4.11 indicate that the BERT model trained using GloVe based weak supervision signals outperform the BERT model trained with relevance signals, showing the effectiveness of the weak supervision strategy which di-

Table 4.11 Comparison of performance between models trained on (question,passage) relevance judgements against the BERT pseudo-labeling model and baseline. The performance is measured based on the clustering performance of the relevant passages of the same type.

Models	P@10	P@20
GloVe baseline	0.0663	0.0460
P-BERT Baseline	0.0838	0.0626
GloVe point-wise model	**0.0964**	**0.0686**
BERT trained with relevance	0.0280	0.0217

rectly uses the kNN clustering signals. The GloVe and BERT baselines also perform better than the relevance trained BERT model, which re-enforces the benefit of using kNN clustering.

Table 4.12 Results on NFPassageQA_Sim dataset for clustering relevant passages (Rel Clusters) for the labels which combine different pseudo-labels.† indicates significance with respect to corresponding baselines. P-BERT refers to pre-trained BERT. The scores for trained model which beats the corresponding baseline has been marked in bold.

Pseudo-labeler	Model	Rel Clusters			
		P@10	P@20	R@10	R@20
LM+GloVe	Baseline	0.1524	0.1129	0.1592	0.2477
	Point-wise	**0.1999**†	**0.1642**†	**0.2412**†	**0.4071**†
LM+P-BERT	Baseline	0.2312	0.1756	0.2713	0.4547
	Point-wise	0.1928	0.1421	0.1935	0.3082
GloVe+P-BERT	Baseline	0.2295	0.1748	0.2627	0.4273
	Point-wise	0.1765	0.1404	0.1993	0.3353
LM+GloVe+P-BERT	Baseline	0.2238	0.1758	0.2539	0.4233
	Point-wise	0.2048	0.1550	0.2086	0.3338

4.4.2.5 Performance of model combinations

Table 4.12 and Table 4.13 reports the results for models trained with the weak labels obtained by combining the similarity scores of the individual models. This experiment was conducted to determine if complementary information from different sources, combined linearly could improve the performance on the task. The combina-

tion model, LM+GloVe improves over the baseline, which indicates the effectiveness of using GloVe signals. Overall, the model performs comparably to the GloVe models.

Table 4.13 Results on NFPassageQA_Sim dataset for clustering relevant passages of the same type (Sim Clusters) for the labels which combine different pseudo-labels.† indicates significance with respect to corresponding baselines. P-BERT refers to pre-trained BERT. The scores for trained model which beats the corresponding baseline has been marked in bold.

Pseudo-labeler	Model	Sim Clusters			
		P@10	P@20	R@10	R@20
LM+GloVe	Baseline	0.0700	0.0464	0.1448	0.1807
	Point-wise	**0.0893**†	**0.0666**†	**0.1719**	**0.2621**†
LM+P-BERT	Baseline	0.0858	0.0643	0.1745	0.2499
	Point-wise	0.0836	0.0556	0.1601	0.2106
GloVe+P-BERT	Baseline	0.0903	0.0638	0.1873	0.2480
	Point-wise	0.0823	0.0585	0.1609	0.2206
LM+GloVe+P-BERT	Baseline	0.0883	0.0659	0.1784	0.2531
	Point-wise	0.0841	0.0584	0.1624	0.2183

4.5 Summary

In this chapter, we describe various representations and training schemes used to create fine-grained, overlapping answer similarity clusters, and evaluate how well relevant answer clusters capture answers corresponding to the same answer types, using the NFPassageQA_Sim dataset. We employ kNN clustering with pre-trained representations such as LM, GloVe and BERT, and weak supervision models trained using various pseudo-labels within a BERT framework. We conclude that BERT based weakly supervised models using GloVe pseudo-labels perform the best in generating relevant answer passage clusters covering same answer types and other relevant answers. This result is also consistent with the best performing unsupervised representation, which is the linear combination of GloVe and BERT. Further analysis showed that this behavior was due to the two representations being the same type (ie

semantic, distributed), which tends to retrieve similar or same passages at the top. Since GloVe provides additional complementary information in addition to BERT, in the form of word based semantic information, the combination of the two has an additive effect, ranking the passages with similarity with respect to both these representations higher than each individual case.

In contrast to most QA models which focus only on displaying relevant answers, these answer clusters could be useful in designing more effective QA systems, especially for multi-answer questions to identify different answer options and displaying them to the users. Besides this, since these clusters also tend to cluster relevant answers together, this could also provide an alternative solution to finding more relevant answers.

CHAPTER 5

ANSWER PASSAGE DIVERSIFICATION

5.1 Introduction

In the previous chapter, the focus was on creating different types of clusters and evaluating them on the basis of their performance on the answer passage similarity task. In this chapter, we discuss how these clusters can be used with a diversification model to improve answer diversity and relevance ranking for questions.

Significant research has been done in the area of Search Result Diversification of documents. This has also been covered under TREC Web Track's diversity tasks [12, 88]. An example TREC query (topic 111) from TREC Web Track 2011 is shown in Table 5.1. These models are aimed at queries which are short and ambiguous, for which the information need is not fully specified. For example, the query in Table 5.1 could cover different topics such as "causes", "symptoms" and "treatments" of lymphoma and the main goal of the diversification model is to retrieve documents which cover all these subtopics. For such tasks, term clustering based models [22] have been proposed to create diversified document lists with standard algorithms such as xQUAD [83] and PM-2 [23].

Table 5.1 TREC query topic 111 from TREC Web Track 2011

TREC query	lymphoma in dogs
Subtopic 1	What treatments are available for dogs diagnosed with lymphoma?
Subtopic 2	What are the symptoms of lymphoma in dogs?
Subtopic 3	What are the risk factors or causes of lymphoma in dogs?

However, the focus of our work is specific questions with multiple possible answers, which potentially cover only a single subtopic. For instance for the question "`What are the symptoms of lymphoma in dogs?`", the subtopic is "symptoms" and the diversified answer list should cover all symptoms of the disease. Applying term clusters for this task has been found to be ineffective [91], since it does not provide fine grained subtopics required for generating a diversified list of answers.

Another work which is related to answer diversity ranking is by Omari et al. [65] who used propositional phrase clusters to improve answer diversity, by re-ranking answers corresponding to a question. However, this is a very limited setting, where only a small set of candidate answers for a question is re-ranked.

In this work, we follow the same setting used in passage/document ranking [23, 83, 22] models, where a set of top retrieved answers from a collection are re-ranked. Instead of using term clusters, passage level clusters are used as input to improve answer passage diversity. We demonstrate the efficacy of using different cluster types described in Chapter 4 for the diversification task using the newly created NFPassageQA_Div dataset.

5.2 Answer Passage Diversification Task Definition

Figure 5.1 illustrates the high-level architecture flow, where the answer passage similarity clusters are used as input to a diversification model to generate a re-ranked list of answers. The answer passage clustering task setting is described in Chapter 4, where the nearest neighbors corresponding to each passage are ranked based on a similarity score. The generated passage clusters are then input into a diversified model where an initial ranked list R generated for a question q is diversified based on the clusters.

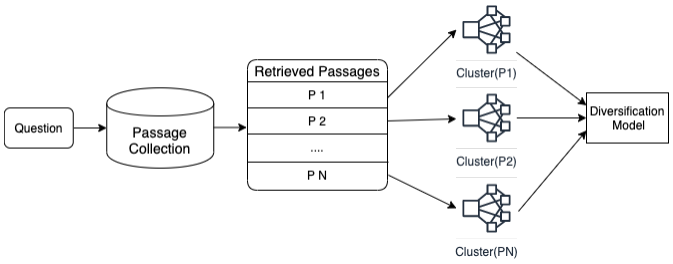


Figure 5.1 Passage Similarity and Diversity pipeline

5.3 Answer Passage Diversification

Answer passage clustering models help in grouping similar passages together. However, the final aim is to be able to display different answer types to the users. Diversification models can capture this information, since they combine relevance and diversity during re-ranking. In this thesis, we use an extension of the Implicit Diversification model MMR (Maximal Marginal Relevance) [9] to diversify the answers. Given a question q, an initial ranked list $R = \{p_1, p_2, ...p_n\}$ of answer passages generated using a standard retrieval model, S representing the ranked list of diversified answers, $f(p_i, p_j)$ the score between passages p_i and p_j, $Clus_m(p)$ the m most similar passages to passage p, p^* the answer passage selected at each step of ranking, the standard MMR model is defined as follows:

$$p^* = \underset{p_i \in (R-S)}{\operatorname{argmax}} (1-\delta) rel(p_i, q) + \delta \max_{p_j \in S} f(p_i, p_j) \tag{5.1}$$

We modify this for cases where the passage p_j in S is ranked within the top 10 of the ranked list R (i.e highly relevant to the question). Here, the diversity component $\max_{p_j \in S} f(p_i, p_j)$ is replaced by $\max_{p_j \in S} \max_{p_k \in Clus_m(p_j)} f(p_i, p_k)$.

For these cases, instead of finding the maximum similarity between an element in R and passage p_j in S, we consider the maximum similarity of p_j with top m most similar cluster elements with respect to p_j. We only expand passages within S which are highly ranked with respect to the question to limit the noise which could be

introduced by the non-relevant passages. We also experimented with other settings such as using cluster elements in R and found this setting to be the best. We will call this `MMR Cluster` to distinguish it from the other variants.

5.4 EXPERIMENTAL SETUP

5.4.1 Data

In order to evaluate the output generated by diversified model, we used the newly generated dataset NFPassageQA_Div described in Chapter 3.

5.4.2 Baselines

The initial retrieval run is obtained using the Query Likelihood model [18]. The diversity re-ranking is performed over the top 100 retrieved answers. We consider a number of standard baselines and compare against them.

5.4.2.1 Query Likelihood (QL)

This is the initial retrieval run generated using default Dirichlet prior smoothing (μ=2500).

5.4.2.2 Maximal Marginal Relevance (MMR)

This is the classic MMR implementation [9], which uses a greedy implicit diversification algorithm to generate a diversified output. We use two different versions of this as baseline: MMR Sparse and MMR P-BERT. MMR Sparse is the classical IR approach using a sparse vector representation for terms, where the different dimensions contain term frequency information. MMR P-BERT uses the Layer-1 ([CLS]) BERT representation for the passages.

5.4.2.3 Term Level Diversification

This was introduced by Dang et al. [22] and is an explicit diversification model, where a set of topic terms are first generated by an algorithm called DSPApprox and these are then used with xQUAD and PM-2 algorithms to generate a final list. We display results for xQUAD and omit PM-2, as xQUAD consistently outperformed the PM-2 model for this dataset. This is also consistent with the findings in [91].

5.4.3 Implementation Details

The parameters for the diversity baselines are set by cross-validation. $f(p_i, p_j)$ corresponds to the euclidean distance between passages p_i, p_j for BERT and GloVe based representations, for LM this corresponds to $\frac{1}{1+sim}$ where sim is the cross-entropy similarity value discussed in Chapter 4. δ value for the `MMR Cluster` approach is set to 0.5 and parameter m is set to 40 for WS GloVe model and 60 for P-BERT to reflect the best performance in each case. In order to maintain consistency with our approach, topic terms for the term-level diversification baseline are generated from top 200 retrieved set using the Query Likelihood baseline model, which is the same setting used for answer passage clustering.

5.4.4 Evaluation

To evaluate diversification models, standard diversity metrics such as Precision-IA@k, S-Recall@k and α-NDCG@k are used. We also measure relevance values using Precision@k, Recall@k and NDCG@k metrics. Here k is set to 10. Statistical significance is measured using the paired two-tailed t-test with p-value < 0.05.

5.5 Results and Analysis

Tables 5.2 and 5.3 shows the diversity and relevance results respectively on various diversity models for the NFPassageQA_Div dataset. In general, Term Level Diversification model using xQUAD is a competitive baseline outperforming both the QL

Table 5.2 Diversity Results on NFPassageQA_Div dataset for different diversification methods. Q, S, T, B indicates significance with respect to the baselines QL, MMR Sparse, TLD and MMR P-BERT respectively. Here TLD refers to Term level Diversification [22]. P-BERT refer to the unsupervised model while WS GloVe refer to the weak supervision model trained with GloVe signals. The scores for the best performing model has been marked in bold.

Type	Model	Diversity		
		Prec-IA	S-Recall	α-NDCG
Baselines	QL	0.2104	0.6905	0.4671
	MMR Sparse	0.0711	0.4743	0.2996
	TLD xQUAD	0.2166	0.7057	0.4705
	MMR P-BERT	0.1732	0.6118	0.4083
Cluster	MMR Cluster GloVe	$0.2117^{S,T,B}$	$00.7012^{S,B}$	$0.4667^{S,T,B}$
	MMR Cluster P-BERT	$0.2290^{Q,S,T,B}$	$0.7317^{Q,S,B}$	$0.4867^{Q,S,T,B}$
	MMR Cluster GloVe+P-BERT	$0.2293^{Q,S,B}$	$0.7333^{Q,S,B}$	$0.4810^{S,B}$
	MMR Cluster WS GloVe	$\mathbf{0.2344}^{Q,S,T,B}$	$\mathbf{0.7530}^{Q,S,T,B}$	$\mathbf{0.4939}^{Q,S,T,B}$

baseline and MMR with standard BERT representation. Clusters from unsupervised BERT representations and weak supervision based cluster performs well across all diversity and relevance metrics. The cluster generated using GloVe weak labels (WS GloVe) performs the best, significantly beating all the baselines.

5.5.1 Impact of using clustering for diversity

In this section, we investigate how clustering helps in retrieving higher quality answers for the question answering system. To this end, we first analyze the Win/Tie/Loss statistics for the top performing baseline and clustering model with respect to S-Recall as given in Table 5.4. Since the dataset consists of questions with single and multiple answer types, we measure this for two cases - all questions and questions with multiple answer types (> 1). Since S-Recall is a metric which measures how well a system discovers new subtopics, this would be relevant only for the second case. As seen in the table, the clustering technique retrieves substantially more answer types than the baseline. We also investigate how clustering contributes to these improvements and found two main reasons for this : (a) In 6 out of 9

Table 5.3 Relevance Results on NFPassageQA_Div dataset for different diversification methods. Q, S, T, B indicates significance with respect to the baselines QL, MMR Sparse, TLD and MMR P-BERT respectively. Here TLD refers to Term level Diversification [22]. P-BERT refer to the unsupervised models while WS GloVe refer to the weak supervision model trained with GloVe signals. The scores for the best performing model has been marked in bold.

Type	Model	Relevance		
		Prec	Recall	NDCG
Baselines	QL	0.3182	0.1043	0.3435
	MMR Sparse	0.1290	0.0404	0.1807
	TLD xQUAD	0.3301	0.1079	0.3503
	MMR P-BERT	0.2763	0.0891	0.2996
Cluster	MMR Cluster GloVe	$0.3204^{S,T,B}$	$0.1050^{S,B}$	$0.3478^{S,T,B}$
	MMR Cluster P-BERT	$0.3451^{Q,S,T,B}$	$0.1130^{Q,S,B}$	$0.3646^{Q,S,T,B}$
	MMR Cluster GloVe+P-BERT	$0.3419^{Q,S,B}$	$0.1118^{Q,S,B}$	$0.3607^{Q,S,B}$
	MMR Cluster WS GloVe	$\mathbf{0.3569}^{Q,S,T,B}$	$\mathbf{0.1162}^{Q,S,T,B}$	$\mathbf{0.3723}^{Q,S,T,B}$

Table 5.4 Win/Tie/Loss statistics for models compared with the QL baseline with respect to various metrics.

Metric	Models	W/T/L All Questions	W/T/L Multi-Answer Questions
S-Recall	TLD xQUAD	3/87/3	2/49/3
	MMR Cluster WS GloVe	**11/80/2**	**9/43/2**
Prec-IA	TLD xQUAD	11/76/6	6/44/4
	MMR Cluster WS GloVe	**30/59/4**	**17/34/3**
α-NDCG	TLD xQUAD	25/45/23	16/24/14
	MMR Cluster WS GloVe	**44/37/12**	**28/18/8**

cases, it was found that the improvement in answer type S-Recall was caused by the presence of the currently selected passage (using MMR) within the cluster of an already retrieved relevant passage in set S. For example, for the question "`How do you prevent chicken from drying out when you cook it?`" (with 5 answer types), an already relevant selected passage in set S, containing 2 answer types "`sprinkle water, wrap in foil`" has another answer with answer type "`coat chicken and fry in oil`" in its cluster and the similarity score for this would be higher than other

passages and is hence retrieved. This demonstrates how "Rel Clusters" or clustering relevant passages correlates with improvement in this metric. (b) In the remaining 3 out of 9 cases, we found that non-relevant passage was responsible for retrieving relevant passage due to its presence in its cluster. These passages though non-relevant, had some contextual similarity to the expected answers. For example, for the question "`How do I get rid of mice humanely?`", a non-relevant passage "`Ask them to leave politely`", had a relevant answer with answer type "`Use live traps`" within its cluster, which was subsequently retrieved by the algorithm.

We also studied the behavior with respect to the Precision-IA metric, which measures the average number of relevant passages retrieved for each answer type. This metric would be pertinent for both cases where questions have a single answer type and for those with multiple answer types. We found the behaviour similar to that of S-Recall. Out of the 30 questions which improved compared to the QL baseline as given in Table 5.4, the gains for 22 of these can be attributed to their presence in the clusters of relevant passages present in set S. This demonstrates how "Sim Clusters" or clustering relevant passages of the same type correlates with improvement of the Prec-IA metric. The improvements in the remaining 7 questions, are due to the presence of relevant passages in clusters of non-relevant passages same as in the S-Recall metric.

The combination of retrieving more answer types and relevant passages for each answer type contributes to the improvement in α-NDCG. Significant gains in α-NDCG metric also demonstrates that more relevant passages are ranked higher. This also directly correlates with the improvement in various relevance metrics such as NDCG, Precision and Recall as shown in Table 5.3.

5.5.2 Impact of size of the cluster

In order to get the best performance from the diversity model, we need to pick the right number of top similar elements (m) or cluster size. To that end, we study how α-NDCG value of the models changes with increase in m. As shown in Figure 5.2, we plot two different clusters generated using P-BERT representations and weakly labeled GloVe with BERT against different values of $m = \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$. For both cases, we observe that the efficacy of the model decreases after a threshold. This was also observed with S-Recall and Precision-IA metrics. This behavior can be attributed to the additional noise introduced by less similar passages, which is added as we increase the cluster size.

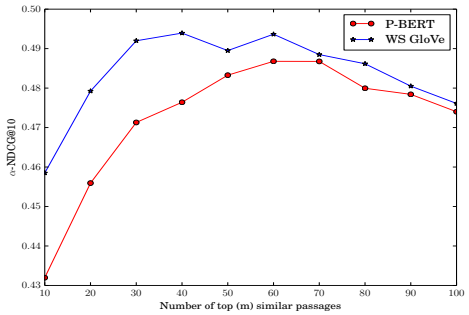


Figure 5.2 α-NDCG across Cluster size

5.5.3 Performance Comparison with Term Level Diversification baseline

We qualitatively compared the answers retrieved by the Term Level Diversification baseline, which is a high-performing model for document diversity and the cluster based MMR models and observed that the terms used for diversification in the term based model is insufficient to differentiate between non-relevant and relevant answers. For example, for the question "`How to get rid of warts?`, some of the top terms used by the term based model are "`remove, try, tape, work, freeze`". While some of these terms do refer to methods for wart removal such as "using tape" or

"freeze", this also retrieved other non-relevant passages with the same terms. This issue is mitigated to a large extent in cluster based models due to the contextual information captured by them.

5.5.4 Performance comparison between different clustering models

In order to compare the performance of various clustering models in combination with MMR, we performed the experiments with clusters of size $m = \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$ and reported the maximum value in Table 5.5. The weakly supervised model performed the best amongst all the models with the unsupervised GloVe model performing the worst. This shows the efficacy of using weak supervised clusters as opposed to clusters generated using unsupervised representations.

Table 5.5 α-NDCG metric comparison for MMR models using different clustering models

Clustering Models	α-NDCG
MMR Cluster Glove	0.4667
MMR Cluster P-BERT	0.4867
MMR Cluster WS Glove	**0.4939**

5.6 Summary

In this chapter, we focus on applying answer passage clustering models described in Chapter 4, to create diversified answer rankings. We experiment using these clusters within a diversification framework such as MMR and conclude that they can be used to generate diverse and relevant outputs. We evaluate them using the NFPassage_Div dataset described in Chapter 3, and show that they perform significantly better than standard baselines using both diversity and relevance metrics. Further analysis showed that the effectiveness of passage clusters lies in its ability to identify other relevant passages, which could then be considered as candidates in the diversification process. Besides, the passages retrieved via this method is more effective

than the ones retrieved using term based methods such as Term level Diversification and MMR Sparse. The results also indicate that quality of clusters have a direct impact on the performance of the model, with the best clustering model (WS GloVe) performing the best. These results demonstrate the efficacy of combining answer passage clustering with diversification models to generate high quality answer rankings, which would be very useful in QA systems which has multi-answer questions with overlapping answers and even for questions with multiple relevant answers belonging to the same type.

CHAPTER 6

CLUSTER MODELS IN CONVERSATIONAL SEARCH

In previous chapters, we studied grouping or clustering of answers for multi-answer questions and how these could be used to improve answer quality by incorporating them into a diversification model. In this chapter, we explore how to apply such models to the conversation search task from the perspective of using the answer passage clusters to identify interrelated questions within a conversation and using them to improve the passage retrieval task.

Recent work in Conversational Search addresses different types of information needs such as response ranking [29, 102, 103], conversational question answering [11, 82], next question prediction and document retrieval [1], user intent prediction [73] etc. While the response ranking systems display re-ranked sets of candidate answers based on conversational context, other models such as conversational question answering identifies answer spans within text passages in a machine comprehension setting [74, 75] or an open-retrieval setting [72, 71]. Models using clarification questions [1] to address broad user needs is yet another area of research, where various query facets could be used to drill down to specific user information needs.

In this chapter, we specifically address the task of passage retrieval in a conversational setting i.e, given a current query and previous query turns, our goal is to retrieve passages which satisfy user information needs. Since queries within a conversation are often under-specified, with references to previous query turns, this task is often difficult. A commonly used solution is create fully grown queries using previous query turns or relevant prior responses. To address this task of generating

de-contextualized queries, many solutions have been proposed. These include query expansion methods where queries are reformulated by selecting terms from previous turns to add to the current turn [93, 56]. Another area of research is query rewrite models, where data annotation is first performed to collect query-rewrite information [28, 3], which is then used to fine-tune large transformer models such as T5 [55]. Some other techniques have also been employed such as few-shot learning approaches using dense retrieval models trained using a teacher-student framework [105], and zero-shot language models using GPT-2 [76]. Most of these models depend on previous query turns and responses to reformulate the current query. These responses are the highest ranked passages retrieved by the system. In this chapter, we study if the inter-relationships between queries in combination with passage clusters generated using top retrieved passages could be used to generate better history response candidates to improve the passage retrieval task.

Table 6.1 TREC CAsT 2020 Topic 89

Turn Id	Manual Utterance
4	Where is the Venus flytrap native to? *The Venus flytrap is native to North and South Carolina*
5	How do Venus flytraps attract and catch their prey?
6	What other carnivorous plants are native to North and South Carolina?

In order to identify such responses, we have to first identify the query most correlated with the current query. Correlation between two queries is based on the extent to which they address similar information needs. In most cases, this would be the query directly preceding the current query in a conversation. However, as shown in Table 6.1, this is not always the case. The example given in the table is an excerpt from Topic 89 in the TREC Conversational Assistance Track (CAsT) [21] 2020 test set. In this example, Turn id 4 could be considered more correlated to Turn id 6 than to 5, since they both refer to the "`native area of plant occurrence`" and

the response to Turn id 4 which is italicized also indicates that the answers to both could refer to "`carnivorous plants native to a particular area`". Also, under a normal setting, we use the top response from the previous (or history) query as the history response candidate. Since these queries are related, and their responses are also correlated, we could potentially identify alternative history response candidates for a question by considering responses related to both. Since passage clustering provides a means of identifying alternative or similar responses, using them in conjunction with a retrieval model provides a means of achieving this goal. Since these alternative candidates are related to both prior (or history) and current query, it has a higher likelihood of adding more relevant information which could be useful to rewrite the current query in more useful and meaningful ways. An overview of the basic pipeline is shown in Figure 6.1.

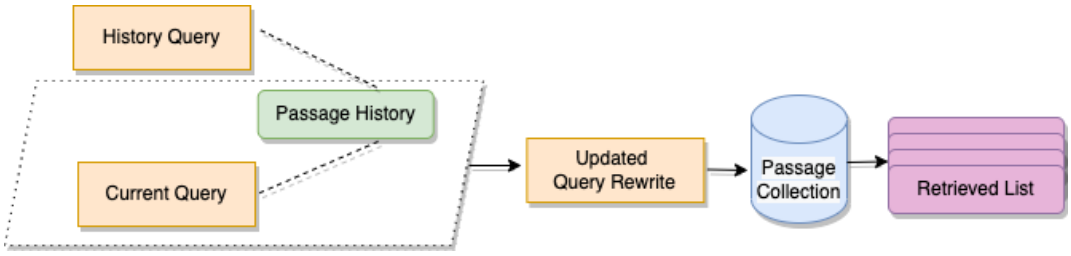


Figure 6.1 Basic pipeline

Therefore, in this work we leverage the idea that correlated queries could have responses which contain related information and could serve as better contexts than the default top response. Currently, there are no methods which directly address the problem of improving the context responses by looking at related responses between interrelated queries. We first propose a method to use kNN passage clusters generated from the passages retrieved using baseline query rewrites using a standard retrieval model, to identify response passage candidates. Next, we introduce `RRFCluster`, a Reciprocal Rank Fusion (RRF) based method to identify the best context passage from the candidate set and demonstrate how this information could be given as input

to the rewriter to create better query rewrites than the default top option. Using TREC CAsT 2020 and 2019 datasets, we show the relative performance of this method across different settings for both datasets.

6.1 Task Definition

In this section, we formally define the end to end conversational search task. In TREC Conversational Assistance Track (CAsT) [21, 20], the task has been designed in terms of passage ranking i.e, given a sequence of queries $\{q_{1:i-1}, q_i\}$ within a conversation, the goal is to retrieve a set of relevant passages for every query q_i from a given collection P. We adopt the same basic definition and augment it to include query rewrites generated using a standard rewriter. More formally, the task can be defined as follows: Given a sequence of queries $\{q_{1:i-1}, q_i\}$ in a conversation, with their corresponding rewrites $\{r_{1:i-1}, r_i\}$ generated using a fixed standard rewriter, the goal is to retrieve relevant passages for every query q_i from a collection C. Here, each rewrite r_i is generated by using the entire query history $\{q_{1:i-1}, q_i\}$ as input to the rewriter.

6.2 Methodology

In this section, we describe the process of using the baseline query rewrites $\{r_{1:i-1}, r_i\}$ within a conversation to identify good history passages, which are then used for the passage ranking task. The first subsection describes the various modules in the setup and the second illustrates how these can be combined together to achieve the end results.

6.2.1 Basic Modules

The pipeline consists of three main modules: Rewriter, Ranker (Full Ranker, Reranker) and Passage Clustering. The rewriter is trained on the CANARD dataset

[28] and can take queries as well as longer sequences such as sentences or passages as input. This outputs fully-formed de-contextualized queries which can be used in the subsequent retrieval steps.

The initial full ranking is performed using a BM25 model, and the top 1000 retrieved passages are further re-ranked using a BERT [25] model, trained using the MS MARCO [4] passage ranking dataset.

The re-ranked BERT baseline is used as input to the passage clustering module, where kNN clustering is performed for every passage ranked within top 1000 for each query. Here, k is set to 1000 and euclidean distance is used as the distance metric. During cluster generation, query-passage information representated using distributional models such as BERT [25] and GloVe [92] have been shown to add more context and perform better than passage representations. We adopt identical settings and use BERT based unsupervised representations for our experiments. These representations can be generated using BERT by passing them as inputs to the pre-trained model and taking the $[CLS]$ token embeddings as output.

6.2.2 History Passage Selection

We next investigate how passage clustering information could be helpful in identifying correlated queries within a conversation. We then employ this method to identify reasonable history passages.

6.2.2.1 Query Rewrites Correlation:

For each rewritten question r_i, we select clusters generated for top m_i retrieved passages i.e., for each of the top m_i passages, we select the most similar n_i passages. This results in a set of at most $m_i * n_i$ passages for every r_i. Next, for every query rewrite pair (r_i, r_j) where $i < j$, we identify the set of common passages C_{ij}. The number of passages within this set $|C_{ij}|$ gives a rough indication of the measure of

correlation between the two rewrites. A higher number of common passages between (r_i, r_j), indicates that r_i is likely to be highly correlated with r_j.

6.2.2.2 History selection:

Since the set C_{ij} contains passages with information potentially relevant to both r_i and r_j, we consider this to be the candidate set for history passage selection. To select the best passage from this set, we apply a modified version of Reciprocal Rank Fusion (RRF) [17]. RRF was originally introduced to combine the relevance scores of passages retrieved using different rankers for a particular query. We modify this definition for our task and call it `RRFCluster`. Instead of fusing rankings generated for a single query from different sources, we fuse rankings for the common passages C_{ij} for a pair of rewritten queries (r_i, r_j) and select the passage with the highest RRF score to be the best history passage. During fusion, instead of considering the original ranks of the passage in (r_i, r_j), we first sort them based on the original ranks and use the sorted order to define the new ranks.

More formally, given the updated ranks $(rank(r_i), rank(r_j))$ corresponding to a pair of rewrites (r_i, r_j) for a passage p in set C_{ij}, we define `RRFCluster` score for p as follows.

$$RRFCluster_{score}(p \in C_{ij}) = \frac{1}{t + (rank(r_i)} + \frac{1}{t + (rank(r_j)} \tag{6.1}$$

6.2.3 Adding History information

Once the history passage is identified, the next step is to use this information to generate an updated query rewrite. For this purpose, we extract the first sentence from each history passage. We use only the first sentence, since the BERT reranker has an input length limitation and any information beyond the first sentence could potentially be unhelpful in retrieving more relevant passages. The overall method is shown in Figure 6.2.

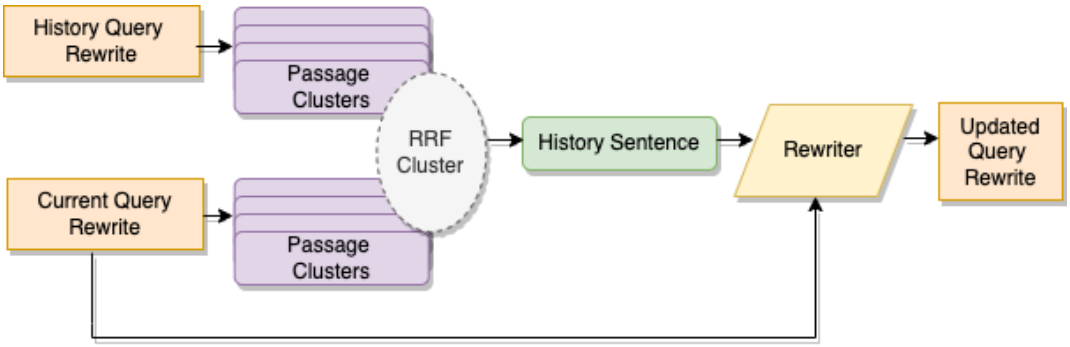


Figure 6.2 Method Overview

So far we have described a generic method to identify history sentence for any pair of rewrites (r_i, r_j) where $i < j$. We need to set the value of the previous turn i for any r_j to incorporate the correct information. To this end, we experiment with the following settings:

- **Prior**[***Input***]: Prior History where $i = j - 1$.
- **Max**[***Input***]: Max History where $i = \text{argmax}_{i \in 1,\ldots,j-1} |C_{ij}|$, i.e, the rewrite at previous turn i which has maximum common passages with the rewrite at current turn j.
- **Comb**[***Input***]: Combination of Prior and Max history passages, where we first include prior history passages and then backtrack to max history passages for the remaining queries for which responses are present.
- **Comb**[***Input***]+**PQ**: Combination of Prior and Max history passages for queries with responses and for rest of the queries, use the query rewrite corresponding to the previous turn as history.
- **Comb**[***Input***]+**R**($\overline{PQ}$): Combination of Prior and Max history passages for queries with responses and for rest of the queries, use the query history passage corresponding to the previous turn as history (if present). The overscore over PQ

$\overline{PQ}$, indicates that we input the responses with the rewrite which combines current and previous query turn rewrites information.

Here, the ***Input*** to the rewriter is the history sentence s_i and the Current Rewrite r_j at turn j [S+R]. We then make up to two passes through the rewriter. We use rewrites generated in a piecemeal fashion as input, instead of adding all the query and responses as input in a single step because the rewriter has difficulty incorporating long range information. We also experiment by combining all rewrites output after incoporating the history information (A(Comb($\overline{S+R}$))).

With the updated rewrite incorporating the history sentence information, we perform another BM25 retrieval from the collection and re-rank using the BERT model. For cases where there is no identified history sentence or queries, we backtrack to the baseline rewrite.

Table 6.2 Data Statistics

Dataset	#Conv	#Queries	#Avg Queries per Conv	Collection (MS MARCO+CAR)
CAsT 2019	50	479	9.6	8.6M+29.7M
CAsT 2020	25	216	8.6	8.6M+29.7M

6.3 Experimental Setup

6.3.1 Data Overview

We used the standard TREC Conversational Assistance Track (CAsT) [1] 2020 and 2019 test sets for evaluation. The data statistics are given in Table 6.2. TREC CAsT 2020 test set consists of 25 conversations with 216 questions. These were questions created by the task organizers to mimic real world conversations. Passages from MS MARCO [4] and TREC Complex Answer Retrieval Paragraph Collection (CAR)

[1] https://www.treccast.ai/

[26] collections were used. De-duplication was also performed on the MS MARCO collection to remove any redundant passages. TREC style pooling was conducted to generate a good result set followed by relevance assessments made by NIST. Five levels of graded judgements were provided by the assessors ranging from $0-4$ with judgements ≥ 2 considered as relevant. We used the Automatic category of the track, using only the raw queries. The TREC CAsT 2019 test set consists of 50 conversations with 479 questions. It uses the same collection and guidelines for creating relevance annotations as the 2020 dataset. The main difference between the two is that in 2020, queries at subsequent turns can depend on previous query turns as well as system responses, while in 2019, it only depends on queries at previous turns. We did not evaluate using the more recent test set TREC CAsT 2021, since we did not participate in the task and do not have full access to it.

6.3.2 Implementation Details

In order to perform BM25 retrieval, we used Pyserini[2] toolkit [53] with parameter settings of $k_1 = 0.82$ and $b = 0.68$[3], which has been found to work well with MS MARCO data for the passage retrieval task. For the baseline BERT model, we used the pre-trained multilingual BERT model[4] integrated with Huggingface transformer tensorflow code, where input length set is to 128. T5-base [77] transformers finetuned using the CANARD dataset [28] is used as the rewriting model. This was trained on a single GeForce GTX 1080 GPU with the input tokens per batch set to 131072 and number of training steps as 1004000. During inference, the beam size is set to 1. For the BERT based clustering experiments [25], the final layer hidden vectors generated

[2]https://github.com/castorini/pyserini

[3]https://github.com/castorini/pyserini/blob/master/docs/experiments-msmarco-passage.md

[4]https://huggingface.co/amberoad/bert-multilingual-passage-reranking-msmarco

using BERT-Large (Uncased) pre-trained model[5] were used. The kNN clustering experiments were conducted using the sklearn toolkit. The hyperparameter t in the RRFCluster model is set to the default value of 60. Corresponding to current rewrite r_j, values for m_j, n_j is selected by performing a grid search and reporting the best value. For the 2020 test set, we set $m_j = 10, n_j = 35$ with the constraint that $|C_{ij}| > 1$ while using prior turn as history and $m_j = 12, n_j = 40$ for history turns with maximum passage candidate overlap. For the 2020 test set, we experiment by setting different values for m and n for history and current turns. For a pair of rewrites (r_i, r_j), $m_i = 4, n_i = 50$ for history turn r_i and $m_j = 1, n_j = 40$ with the constraint that $|C_{ij}| > 1$ for current turn r_j, while considering r_i to be the previous turn. When r_i is the max history, then $m_i = 3, n_i = 50$ and $m_j = 20, n_j = 50$. For the GloVe based experiments, we used 300d pre-trained GloVe [68] vectors[6].

6.3.3 Baselines

We consider the following baselines for our experiments:

- **BM25 model:** This is the initial retrieval run for the queries obtained by using the rewrites $r_{1:i}$, generated by using raw queries $q_{1:i}$ as input to the rewriter model. We use the BM25 parameter settings $k_1 = 0.82$ and $b = 0.68$.

- **BERT-Large:** This is the BERT model fine-tuned using MS Marco passage ranking dataset used as the re-ranker. This also takes the same rewrites as the BM25 model.

- **Prior Top 1:** The rewrites are generated by using the query rewrites $r_{1:i}$ and the history sentence from the top retrieved passage corresponding to the previous query rewrites using the BERT-Large re-ranker.

[5]https://github.com/google-research/bert

[6]https://nlp.stanford.edu/projects/glove/

Evaluation : We use the standard evaluation metrics used in TREC CAsT such as NDCG@3, MRR and MAP. Statistical significance is measured using the paired two-tailed t-test with p-value < 0.05.

Table 6.3 Results on CAsT 2020 and 2019 test sets.*,$\diamond$ and † indicates significance with respect to Prior Top 1, BM25 and BERT baselines respectively. Here under the "Input" column, "Q+S" refers to giving both query rewrites and history sentence as input, while "Q" refers to giving only query rewrite as input. Statistical significance is measured using the paired two-tailed t-test with p-value < 0.05.

Type	Input	Method	CAsT 2020			CAsT 2019		
			NDCG	MRR	MAP	NDCG	MRR	MAP
Manual	Q	BM25	0.2435	0.3746	0.1432	0.3134	0.4897	0.2135
		BERT-Large	0.4741	0.6324	0.2250	0.5400	0.7227	0.3200
Baseline	Q	BM25	0.1499	0.2266	0.0830	0.2833	0.4674	0.1787
		BERT-Large	0.3575	0.4847	0.1868	0.5112	0.6921	0.2904
	Q+S	Prior Top 1 [S+R]	0.3571	0.4903	0.1865	0.5056	0.6801	0.2889
Top	Q+S	Current [S+R]	0.3502	0.4757	0.1861	0.5155	0.6942	0.2926
RRF Cluster	Q+S	Prior [S+R]	0.3644$^{\diamond}$	0.5009$^{\diamond}$	0.1897$^{\diamond}$	0.5124$^{\diamond}$	0.6971$^{\diamond}$	0.2933$^{\diamond}$
		Max [S+R]	0.3631$^{\diamond}$	0.4965$^{\diamond}$	0.1893$^{\diamond}$	0.5155$^{\diamond}$	0.6974$^{\diamond}$	0.2915$^{\diamond}$
		Comb [S+R]	0.3696$^{\diamond}$	0.5061$^{\diamond\dagger}$	0.192$^{\diamond\dagger}$	0.5182$^{\diamond}$	0.7019$^{*\diamond}$	0.2934$^{\diamond}$
		Comb [S+R]+PQ	**0.3738**$^{*\diamond}$	**0.511**$^{\diamond\dagger}$	0.1945$^{*\diamond\dagger}$	0.5199$^{\diamond}$	0.7048$^{*\diamond}$	0.2936$^{\diamond}$
		Comb [S+R]+R($\overline{PQ}$)	0.3654$^{\diamond}$	0.5008$^{\diamond}$	0.1929$^{\diamond\dagger}$	**0.5213**$^{*\diamond}$	**0.7067**$^{*\diamond}$	**0.295**$^{*\diamond\dagger}$
		A(Comb[($\overline{S+R}$)])	0.3728$^{*\diamond\dagger}$	0.5058$^{*\diamond\dagger}$	**0.1975**$^{*\diamond\dagger}$	0.5124$^{\diamond}$	0.6972$^{\diamond}$	0.2922$^{\diamond}$

6.4 Results and Analysis

Table 6.3 gives the final results for the 2020 and 2019 CAsT test sets for the passage ranking task. The column "Input" indicates whether the inputs consists of only query rewrite as input or if this consists of query and response sentence. "Manual" corresponds to results for manual queries which are the gold rewrites given by the task organizers. The second row gives the results for the three baselines. The first two rows corresponds to the initial BM25 retrieval model and BERT-Large re-ranking model. The third row corresponds to the baseline result using the top response from the prior query as input. The row corresponding to "RRFCluster" gives the results for the proposed method, corresponding to different types of history information: previous turn as history (Prev History), Max History and the combination of the two, where max history sentence is used if prev history sentence is not available and a

few other variants described in Section 6.2.3. The results indicate that for the 2019 test set, there is a precision-recall tradeoff between the results with respect to the previous turn history and max history. NDCG and MRR is higher for cases where response sentences from previous history turn is used, while MAP is higher with prior query sentences as history. This behavior is not evident with the 2020 dataset, where in general previous query turn shows the best performance. Unsurprisingly, for both test sets, the combination of responses generated using both types of history contexts outperformed the individual variants. We also experimented by combining all the rewrites of the queries as input to the rewriter and observed that this performs well for the 2020 test set. For queries where response candidates are not available, we experiment with the case where we add the previous query and/or response candidate as the history. Adding only prior query as input works better for 2020 test set, while adding response candidates boost the performance of the 2019 test set conversations. In general, using the `RRFCluster` based response candidate performed significantly better than using prior top 1 response across both precision and recall oriented metrics for both 2020 and 2019 test sets. Also, adding the `RRFCluster` candidate outperformed the baseline BERT-Large re-ranker significantly across all metrics for 2020 test set and the MAP metric for 2019 test set. In the following subsections, we first study the impact of various design choices used to arrive at the final step. Even though some of the improvements shown are minor when studied individually, even small changes can have a significant impact on the final result when added as part of the full pipeline.

6.4.1 History sentence from top retrieved passage

We also study if history sentence information derived from the top retrieved passages of the current turn can act as good history contexts, while used as inputs to the model. The values are reported in the row "Top". Using sentences from the top

retrieved passages of current queries performs worser than the baselines for the 2020 test set, while it improved over the baselines for the 2019 test set. The potential reason for the uncommonly high value in this case, could be the performance of the rewriter on this test set, which is very close to that of Manual rewrites. However, this does not perform as well as the RRFCluster model candidates.

6.4.2 Impact of Clustering and RRF

Since we use a combination of kNN clustering and RRF to identify good contexts, we analyze the relative impact of each of them. Table 6.4 shows the results on CAsT 2020 test set. For the setting with "No Clustering", we consider common passages in the top 1000 retrieved sets for prior and current turn and use these as the candidate set. When RRF is not used, the highest ranked passage within the common set, corresponding to the previous turn is used as the best history passage. In this case, we use the previous turn ranking instead of current turn rank, because the re-ranker is trained using prior history. For the cases where clustering is used, for a rewrite r_i we set the $m_i = 10, n_i = 40$. The results shown in Table 6.4 indicate that clustering in combination with RRF performs the best, closely followed by a model with only clustering. RRF only models performs the worst indicating that clustering is very important for the performance irrespective of whether it is combined with RRF, but RRF only works well in conjunction with the clustering model.

Table 6.4 Relative impact of Clustering and RRF for queries in 2020 test set

Method	Type	NDCG@3	MAP
Prior History[S+R]	No Clustering or RRF	0.3548	0.1852
	RRF only	0.3434	0.1812
	Clustering only	0.36	0.1873
	Clustering+RRF	**0.3613**	**0.1886**

6.4.3 Comparison between BERT and GloVe clusters

We also compare the relative performance of the two types of modeling. To generate candidates for 2020 test set, we set m_i=10 and n_i=35 for both previous and current query rewrites. For 2019 test set, m_i=2 and n_i=50 for previous query and m_i=10 and n_i=50 for the current query rewrite. The relative performance of both the models are shown are Table 6.5. BERT clustering consistently outperform the GloVe clustering models for this task, which is expected since BERT models add more powerful semantic information to generate better candidates.

Table 6.5 BERT vs GloVe clustering impact for queries in 2020 and 2019 test sets

Cluster Representation	CAsT 2020		CAsT 2019	
	NDCG@3	MAP	NDCG@3	MAP
GloVe	0.3562	0.1864	0.508	0.2898
BERT-Large	**0.3632**	**0.1895**	**0.5155**	**0.2915**

6.4.4 Sentence selection

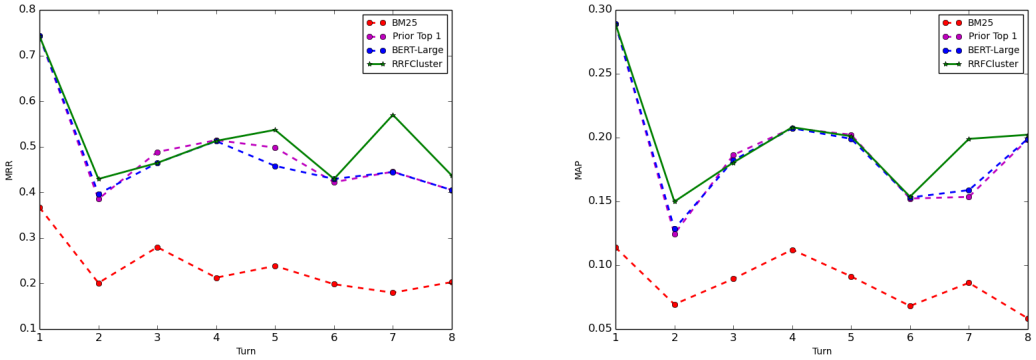
We adopted the heuristic that first sentence of the passage would perform better than others due to the BERT input size limitation. We also compare this with the performance of the sentence within the history passage, which has maximum token overlap with previous query rewrite. As shown in Table 6.6, using the first sentence performed the best. We also experimented with the entire passage and found that using the first sentence as history consistently outperformed all other options across both datasets.

Table 6.6 Relative performance of sentence selection methods in 2020 and 2019 test sets

Sentence Selection Method	CAsT 2020		CAsT 2019	
	NDCG@3	MAP	NDCG@3	MAP
Max Overlap	0.3613	0.1886	0.5128	0.2903
First Sentence	**0.3632**	**0.1895**	**0.5155**	**0.2915**

6.4.5 Conversational Depth Analysis

Next, we perform a depth analysis by measuring the average retrieval performance for each turn. This is shown in Figures 6.3 and 6.4. The results show that the `RRFCluster` in general performs better than the other baselines (indicated by the green solid line in the graph). It also demonstrates the stability of the model, where it consistently stays on or above the baseline for both datasets as opposed to using the Prior Top 1 candidate, which performs worse than the baseline especially in 2019 test set. The results also indicate that the method in general shows consistent performance across all turns, while many of other proposed methods [105, 93] display better performance for the later turns.

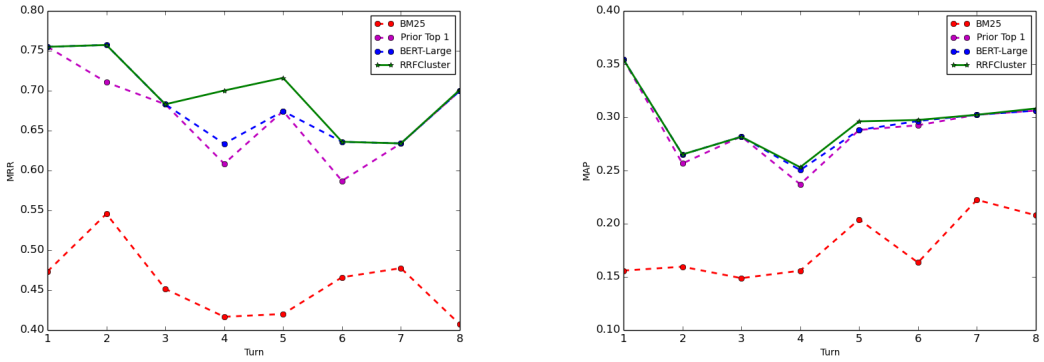


(a) MRR (b) MAP

Figure 6.3 2020 test set retrieval performance across different turns

6.4.6 Qualitative Analysis:

Tables 6.8 and 6.7 illustrate a few examples of how the proposed method improved rewrites in comparison with the baselines with respect to the NDCG@3 metric. As seen from the examples, the `RRFCluster` model adds additional information to the queries from the response passages. The example from Table 6.8 for topic 104 at turn 7 with turn 4 as history illustrates how "`TREC`" acronym was expanded, which



(a) MRR (b) MAP

Figure 6.4 2019 test set retrieval performance across different turns

Table 6.7 History Sentence for CAsT 2019 Topics 59 and 56

Conv Id	Type	Rewrite/History Sent
59_3	Manual	What is the ACL?
	Baseline	What is the ACL?
	Updated	What is the ACL?
	History Sent (`Rank`)	If swelling occurs inside the knee joint after injury, you may have: an **anterior cruciate ligament (ACL)** tear: The ACL is a strong, fibrous band of tissue in the center of the knee. (`427`)
59_4	Manual	What is an injury for the ACL?
	Prior Top 1	What is an injury for the ACL?
	Baseline	What is an injury for the ACL?
	Updated	What is an injury for the **anterior cruciate ligament (ACL)**?
56_4	Manual	How can fossils be used to understand Darwin's theory?
	Baseline	How can fossils be used to understand Darwin's theory?
	Updated	How can fossils be used to understand Darwin's theory?
	History Sent (`Rank`)	Louis Agassiz not only did not accept Charles Darwin's **theory of evolution**, he actively opposed it. (`244`)
56_5	Manual	What is modern evidence for Darwin's theory?
	Prior Top 1	What is modern evidence for Darwin's theory?
	Baseline	What is modern evidence for Darwin's theory?
	Updated	What is modern evidence for Darwin's **theory of evolution**?

improved the performance of the model. A similar example from Table 6.7 for topic 59, turn 4 is the expansion of "`ACL`". Apart from acronym expansion, it is also able

to successfully add other useful information. An example is topic 94 turn 2, where "`1998 Winter Olympics`", the location of the event was correctly added. Similarly in topic 86, turn 7, a specific location in "`Salt Lake City`" was added to make the query more specific. In many of these examples, the additional information is not present in Manual rewrites, but still improves the passage ranking performance. The *Rank* information given as part of the "History Sent" is the rank of the passage corresponding to this sentence with respect to the previous query. This shows that the `RRFCluster` model is able to identify passages at lower ranks which can act as better contexts. This is also indicated by the rewrites of Prior Top 1, which doesn't add useful information or remains identical to the baseline.

6.5 Summary

In this chapter, we study how BERT based passage clusters from Chapter 4 could be used in a conversational search task setting. We study this in the context of identifying alternative context response candidates which could be used to improve the passage ranking task. Using passage clusters generated from baseline query rewrites, where a RRF based method, `RRFCluster` is used to select the best candidate from the history candidate set, we demonstrate the type of settings which would be useful to integrate this information with the rewriter to generate better query rewrites. Experiments show that in general, adding the history sentence information for queries for which this information has been extracted using the model, along with grounding the remaining queries with previous query rewrites or their corresponding responses works the best across both datasets. Results also show that 2020 test dataset performs significantly over the query-only rewrite BERT-Large baseline across all precision (NDCG@3, MRR) and recall oriented metrics (MAP), while this method significantly boost the performance of 2019 dataset with respect to Recall (MAP). We also show that using the new history candidates significantly performs better than using the top

1 retrieved passage from the previous query rewrite across precision and recall oriented metrics in both datasets. These results demonstrate that history passages generated by leveraging the relationships between queries could provide additional information over the top retrieved passage returned by a conversational agent corresponding to the previous query, which would be useful in displaying higher quality answers in a multi-turn conversational task setting. Further analysis indicate that the main advantage of the model is its stability, i.e, the candidates either perform comparably or above that of the baseline query rewrite. This is particularly significant, since the rewrites generated using the wrong responses could potentially worsen retrieval performance significantly with respect to the query-only baseline.

Table 6.8 History Sentence for CAsT 2020 Topics 104, 94 and 86. Here Baseline refers to the rewrites generated using previous raw queries.

Conv Id	Type	Rewrite/History Sent
104_4	Manual	What did Cyril Cleverdon's studies contribute to the evaluation of Information Retrieval systems?
	Baseline	What did the well - known Information Retrieval researchers' studies contribute to evaluation?
	Updated	What did the well - known Information Retrieval researchers' studies contribute to evaluation?
	History Sent (`Rank`)	In 1992, the US Department of Defense along with the National Institute of Standards and Technology (NIST), cosponsored the **Text Retrieval Conference (TREC)** as part of the TIPSTER text program. (`10`)
104_7	Manual	What is TREC for information retrieval research?
	Prior Top 1	What is the TREC?
	Baseline	What is the **TREC**?
	Updated	What is the **Text Retrieval Conference**?
94_1	Manual	How did snowboarding begin?
	Baseline	How did snowboarding begin?
	Updated	How did snowboarding begin?
	History Sent (`Rank`)	Since its mid-1960s inception, ... it is now an event at the Winter Olympics Canadian snowboarder .... won snowboarding's first gold medal at the **1998 Winter Olympics**. (`46`)
94_2	Manual	Interesting. That's later than I expected. Who were the winners of snowboarding events in the 1998 Winter Olympics?
	Prior Top 1	Who were the winners of the 1981 Ski Cooper snowboarding contest?
	Baseline	Who were the winners of the snowboarding?
	Updated	Who were the winners of the snowboarding at the **1998 Winter Olympics**?
86_6	Manual	What are the important non-ski events that happen in Salt Lake City?
	Baseline	What are the important non-ski events that happen in Salt Lake City?
	Updated	What are the important non-ski events that happen in Salt Lake City?
	History Sent (`Rank`)	Winter sports, such as skiing and snowboarding, are popular activities in the **Wasatch Mountains east of Salt Lake City**. (`17`)
86_7	Manual	What are some popular non-winter things to do in the Salt Lake City area?
	Prior Top 1	What about some popular non - winter things to do in the area besides skiing?
	Baseline	What about some popular non - winter things to do in the area besides skiing?
	Updated	What about some popular non-winter things to do in the **Wasatch Mountains east of Salt Lake City** besides skiing?

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Final Remarks

Answer Passage Retrieval is a well studied area of research, with the potential to have a high level of impact in many different applications. This dissertation focuses primarily on multi-answer question answering and proposes solutions to tackle this problem from a different perspective than previously suggested solutions. Our solution focuses on automatically generating answer passage clusters from which answer type information could be automatically generated. We first collect a dataset using manual crowdsourcing with passage pair similarity labels, from which answer types could be identified. Next, we propose a passage clustering task, in order to create answer passage clusters with respect to a question, indicating potential answer types. For this purpose, we experiment with supervised and weakly supervised strategies. Once we have the clusters, we use these as part of a diversification algorithm to improve answer passage ranking for multi-answer questions. Finally, we demonstrate how questions within a conversation could have common related passages in top answers, which could serve as good candidate history passages to improve the conversational search task.

In Chapter 3, we study how to create a manually annotated test set with answer type information. Since, answer type information for questions are not available, the annotation process could not be framed directly. Therefore we perform this as a two step process. First, we label all possible passage pairs (with respect to a question) with similarity labels and then define a process to group these answers into answer

types. For labeling, we employ Mechanical Turk workers, and define the annotation guidelines in detail. Since there is no prior work where annotation of this type has been performed, we perform several pilot runs before finalizing the guidelines. Due to the complexity of the task, we enforce a closed annotation process, with a limited set of annotators to reduce labeling errors. The passage pair similarity labels are then used as the input to automatically generate answer types for questions. This results in the creation of two datasets, NFPassage_Sim with answer passage similarity labels and NFPassage_Div with the answer type information. This work demonstrates how QA datasets with answer type information could be generated via an indirect two-step process.

In Chapter 4, we study how to create passage clusters. We define a kNN based clustering technique, which has been used widely in Cluster based Retrieval, as the basic framework. We explore how clustering using term based statistics differs from distributional representations such as GloVe and BERT. For this study, we first experiment with unsupervised representations, and then explore how these signals could be used with a weak supervision method to improve the similarity task performance. With respect to the performance of unsupervised representations, we study many different input settings across all representations and conclude that distributional representations outperform term-based ones significantly, with BERT based models performing the best. Apart from the individual representations, we also experimented with their linear combinations, where combining GloVe and BERT outperformed all other variants. We also perform a comparative study to determine the types of signals that could be useful as weak supervision signals for training similarity models. A BERT-base model fine-tuned with pointwise and pairwise losses is used as the basic framework. We train this model using three different weak supervision signals: LM (Language Modeling), GloVe and BERT and found the GloVe signals to be the most useful. Even though the BERT pre-trained model has a sufficiently high level of basic

semantic information, using GloVe as weak signal complements BERT, and boosts the performance of the model. Further analysis showed that this was could mainly be attributed to the similarity of the input signals, where both tend to retrieve similar passages at higher ranks, therefore passages which are most similar to both tend to get an even higher similarity score in a combination setting, which explains why the two signals combine better together than with LM. We demonstrate the effectiveness of these models by evaluating them using NFPassage_Sim dataset.

In Chapter 5, we use the passage clusters to generate better answer rankings by using them in a diversification framework `MMRCluster`. We define a simple process of extending the top retrieved passages for questions with their clusters, in order to generate a final diversified answer ranking. We evaluate them using the NFPassage_Div dataset, and show that this method improves the answer ranking quality in comparison to the baselines, using both relevance and diversity based metrics. We also performed further analysis to study why clusters are effective for this task and found the main reason to be its effectiveness in identifying more relevant passages than the baseline models and using them in the diversification framework.

In Chapter 6, we apply these clusters in a conversation search task. Most prior work identified previous queries as history and used these to generate improved query reformulations to retrieve better passages. Many of these models also employ response passages in addition to the query information. Since queries within a conversation are related and by extension, their responses, these relationships could be used to generate better response candidates for improving search. We first identify a candidate set of passages using answer passage clustering and then apply a modified version of RRF, `RRFCluster` to identify the best response candidate. We next define how this could be used with a rewriter to generate better rewrites for the passage ranking task. Using CAsT 2020 and 2019 test sets, we demonstrate that the query rewrites generated using these passages perform better than the top retrieved system

response for the prior query. We also show that this outperforms the query-only rewrite baseline significantly over precision and recall-oriented metrics for 2020 test set and recall-oriented metrics for 2019 test set. Furthermore, the depth performance on conversational task also indicates that this method tends to generate more stable query rewrites than other baselines, which is significant since the method adds additional information from the responses and adding non-relevant information could be detrimental to the search performance.

7.2 Future Work

In this dissertation, we have attempted to fill a gap with respect to studying the inter-relationships between answers and how this could be useful in many different areas. However, there are many extensions which are possible as future work.

7.2.1 Datsasets for multi-answer non factoid questions

We demonstrated how a small high quality dataset could be collected for evaluation of multi answer questions. We adopted a two step process of annotating similarity labels and then converting them into answer groups. However, in order to model and train fully supervised models and leverage the latest model innovations, a large scale dataset would be needed. Since the similarity labeling involves looking at all possible relevant answer pairs for a question, annotation for a large QA dataset using this method would be very expensive. Also, direct annotation of answer types would not be possible due to absence of answer type information. Therefore, more efficient methods to tackle both these problems would need to be investigated.

7.2.2 Answer Passage Clustering

We studied the answer passage clustering task under the framework of Cluster-based IR models. Various unsupervised representations as well as weak supervision settings were studied to identify the best performing models for this task. We adopted

kNN as the clustering algorithm and optimized it to cluster top 200 retrieved passages. More recently, efficient libraries such as FAISS [35] have been developed to perform clustering over billions of dense vectors. Since the clustering enables even lower ranked similar passages to be identified, this could be used to extend the model to generate clusters using the entire passage collection instead of top 200, which could further improve the performance.

We also studied a linear combination method (Chapter 4) to combine signals from different sources such as LM, BERT and GloVe. However, experiments indicate that the linear combination, while effective for signals of the same type such as BERT and GloVe (semantic), it is not useful for combining different types of signals, such as LM (term based) and semantic models. We also observed the same behaviour in a weak supervision setting, where GloVe enhances BERT-base model, but LM doesn't. Therefore, other methods must be studied to understand how these could be combined. One avenue to look at are the hybrid retrieval models, which outperforms the individual variants.

7.2.3 Answer Passage Diversification

In order to solve the answer diversification task, we employ answer passage clusters described in Chapter 4 and show that these are effective in identifying answers which belong to a fine-grained subtopic, which could be used with an implicit model such as MMR to improve answer quality. In Chapter 2, we applied a method to convert ground truth similarity labels in NFPassage_Sim dataset to answer type information. This method could be extended to an unsupervised or weakly supervised setting to explicitly create answer clusters. These clusters could then be used with an explicit diversification model such as xQUAD [83] and PM-2 [23] to diversify answers. These weak labels could also be used to optimize metrics such as α-NDCG to achieve better performance for the diversity task.

Since the aim is to improve answer quality by discovering fine-grained subtopics, another possible avenue is to cast the problem in terms of fair ranking task [7, 27]. The fairness task focuses on ranking documents from the perspective of content providers instead of end users. Since diversity and relevance models focus on the user side, it would be interesting to see if this could generate more useful answers or other alternative answers.

7.2.4 Conversation Search Task

We showed that good response candidates to generate better query rewrites could be identified by leveraging the inter-relationships between queries. This was demonstrated using a strong BERT-Large re-ranker over a sparse BM25 full ranker model as the baseline. The passage clustering was performed over top 1000 retrieved passages to improve efficiency. However, since clustering poses a way to identify low-ranked related passages, a good alternative model to test this would be with a high performing dense retrieval based full ranker such as ANCE (Approximate nearest neighbor Negative Contrastive Estimation) [99]. A dense retriever could retrieve more semantically related passages than BM25, and it would be interesting to see if passage clusters would be able to further enhance the performance due to this factor. Another potential direction is to test this with hybrid sparse-dense retrieval models [52, 54, 95], to study its relative performance with respect to sparse and dense models. Apart from studying the impact of using different baselines, it would also be interesting to apply this to other conversational models such as document ranking tasks [1] and those modeling explicit feedbacks [6].

One of the limitations of our experiments was the lack of processing power to train larger rewriter models. To overcome this, we trained a T5-base model instead of its larger variants as the rewriter. Furthermore, the experiments were conducted by giving the inputs separately to the rewriter to account for this. Extending these

models using the larger models could reveal other more interesting insights, since it can access longer range dependencies and can accept longer inputs.

BIBLIOGRAPHY

[1] Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W Bruce Croft. 2019. Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42nd international acm sigir conference on research and development in information retrieval*. 475–484.

[2] James Allan, Margaret E Connell, W Bruce Croft, Fang-Fang Feng, David Fisher, and Xiaoyan Li. 2000. *Inquery and trec-9*. Technical Report. Massachusetts Univ Amherst Center For Intelligent Information Retrieval.

[3] Raviteja Anantha, Svitlana Vakulenko, Zhucheng Tu, Shayne Longpre, Stephen Pulman, and Srinivas Chappidi. 2021. Open-Domain Question Answering Goes Conversational via Question Rewriting. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 520–534.

[4] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. MS MARCO: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268* (2016).

[5] Nicholas J Belkin, Colleen Cool, Adelheit Stein, and Ulrich Thiel. 1995. Cases, scripts, and information-seeking strategies: On the design of interactive information retrieval systems. *Expert systems with applications* 9, 3 (1995), 379–395.

[6] Keping Bi, Qingyao Ai, and W Bruce Croft. 2021. Asking Clarifying Questions Based on Negative Feedback in Conversational Search. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*. 157–166.

[7] Asia J. Biega, Fernando Diaz, Michael D. Ekstrand, and Sebastian Kohlmeier. 2019. Overview of the TREC 2019 Fair Ranking Track. In *The Twenty-Eighth Text Retrieval Conference (TREC 2019) Proceedings*.

[8] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Empirical Methods in Natural Language Processing*.

[9] Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. 335–336.

[10] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to Answer Open-Domain Questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1870–1879.

[11] Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question Answering in Context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 2174–2184.

[12] Charles L Clarke, Nick Craswell, and Ian Soboroff. 2009. *Overview of the trec 2009 web track*. Technical Report. Waterloo Univ (Ontario).

[13] Daniel Cohen and W Bruce Croft. 2016. End to end long short term memory networks for non-factoid question answering. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval*. 143–146.

[14] Daniel Cohen and W Bruce Croft. 2018. A hybrid embedding approach to noisy answer passage retrieval. In *European Conference on Information Retrieval*. Springer, 127–140.

[15] Daniel Cohen, Scott M Jordan, and W Bruce Croft. 2019. Learning a Better Negative Sampling Policy with Deep Neural Networks for Search. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*. 19–26.

[16] Daniel Cohen, Liu Yang, and W Bruce Croft. 2018. WikiPassageQA: A benchmark collection for research on non-factoid answer passage retrieval. In *The 41st international ACM SIGIR conference on research & development in information retrieval*. 1165–1168.

[17] Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. 758–759.

[18] W Bruce Croft, Donald Metzler, and Trevor Strohman. 2010. *Search engines: Information retrieval in practice*. Vol. 283. Addison-Wesley Reading.

[19] W Bruce Croft and Roger H Thompson. 1987. I3R: A new approach to the design of document retrieval systems. *Journal of the american society for information science* 38, 6 (1987), 389–404.

[20] Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2020. CAsT 2020: The Conversational Assistance Track Overview. In *TREC*.

[21] Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2020. TREC CAsT 2019: The conversational assistance track overview. *arXiv preprint arXiv:2003.13624* (2020).

[22] Van Dang and Bruce W Croft. 2013. Term level search result diversification. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. 603–612.

[23] Van Dang and W Bruce Croft. 2012. Diversity by proportionality: an election-based approach to search result diversification. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. 65–74.

[24] Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W Bruce Croft. 2017. Neural ranking models with weak supervision. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 65–74.

[25] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186.

[26] Laura Dietz, Manisha Verma, Filip Radlinski, and Nick Craswell. 2017. TREC Complex Answer Retrieval Overview. In *TREC*.

[27] Michael D. Ekstrand, Graham. McDonald, Amifa. Raj, and Isaac. Johnson. 2022. Overview of the TREC 2021 Fair Ranking Track. In *The Thirtieth Text Retrieval Conference (TREC 2021) Proceedings*.

[28] Ahmed Elgohary, Denis Peskov, and Jordan Boyd-Graber. 2019. Can You Unpack That? Learning to Rewrite Questions-in-Context. 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)*. 5918–5924.

[29] Jiafeng Guo, Yixing Fan, Liang Pang, Liu Yang, Qingyao Ai, Hamed Zamani, Chen Wu, W Bruce Croft, and Xueqi Cheng. 2020. A deep look into neural ranking models for information retrieval. *Information Processing & Management* 57, 6 (2020), 102067.

[30] Helia Hashemi, Mohammad Aliannejadi, Hamed Zamani, and W Bruce Croft. 2020. ANTIQUE: A non-factoid question answering benchmark. In *European Conference on Information Retrieval*. Springer, 166–173.

[31] Helia Hashemi, Hamed Zamani, and W Bruce Croft. 2020. Guided transformer: Leveraging multiple external sources for representation learning in conversational search. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1131–1140.

[32] Sha Hu, Zhicheng Dou, Xiaojie Wang, Tetsuya Sakai, and Ji-Rong Wen. 2015. Search result diversification based on hierarchical intents. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. 63–72.

[33] Hsin-Yuan Huang, Eunsol Choi, and Wen-tau Yih. 2018. Flowqa: Grasping flow in history for conversational machine comprehension. *arXiv preprint arXiv:1810.06683* (2018).

[34] Nick Jardine and Cornelis Joost van Rijsbergen. 1971. The use of hierarchic clustering in information retrieval. *Information storage and retrieval* 7, 5 (1971), 217–240.

[35] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data* 7, 3 (2019), 535–547.

[36] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, 1601–1611.

[37] Endri Kacupaj, Barshana Banerjee, Kuldeep Singh, and Jens Lehmann. 2021. ParaQA: A Question Answering Dataset with Paraphrase Responses for Single-Turn Conversation. In *European Semantic Web Conference*. Springer, 598–613.

[38] Mostafa Keikha, Jae Hyun Park, and W Bruce Croft. 2014. Evaluating answer passages using summarization measures. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. 963–966.

[39] Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. *Advances in neural information processing systems* 28 (2015).

[40] Kalpesh Krishna, John Wieting, and Mohit Iyyer. 2020. Reformulating Unsupervised Style Transfer as Paraphrase Generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 737–762.

[41] Robert Krovetz. 2000. Viewing morphology as an inference process. *Artificial intelligence* 118, 1-2 (2000), 277–294.

[42] Oren Kurland. 2009. Re-ranking search results using language models of query-specific clusters. *Information Retrieval* 12, 4 (2009), 437–460.

[43] Oren Kurland and Eyal Krikon. 2011. The Opposite of Smoothing: A Language Model Approach to Ranking Query-Specific Document Clusters. *Journal of Artificial Intelligence Research* 41 (2011), 367–395.

[44] Oren Kurland and Lillian Lee. 2004. Corpus structure, language models, and ad hoc information retrieval. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*. 194–201.

[45] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics* 7 (2019), 453–466.

[46] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: a Benchmark for Question Answering Research. *Transactions of the Association of Computational Linguistics* (2019).

[47] Dawn Lawrie, W Bruce Croft, and Arnold Rosenberg. 2001. Finding topic words for hierarchical summarization. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. 349–357.

[48] Dawn J Lawrie and W Bruce Croft. 2003. Generating hierarchical summaries for web searches. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. 457–458.

[49] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*. PMLR, 1188–1196.

[50] Anton Leuski. 2001. Evaluating document clustering for interactive information retrieval. In *Proceedings of the tenth international conference on Information and knowledge management*. 33–40.

[51] Shangsong Liang, Zhaochun Ren, and Maarten De Rijke. 2014. Fusion helps diversification. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. 303–312.

[52] Jimmy Lin and Xueguang Ma. 2021. A few brief notes on deepimpact, coil, and a conceptual framework for information retrieval techniques. *arXiv preprint arXiv:2106.14807* (2021).

[53] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A Python Toolkit for Reproducible Information Retrieval Research with Sparse and Dense Representations. In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*. 2356–2362.

[54] Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2020. Distilling dense representations for ranking using tightly-coupled teachers. *arXiv preprint arXiv:2010.11386* (2020).

[55] Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2020. TREC 2020 Notebook: CAsT Track.. In *TREC*.

[56] Sheng-Chieh Lin, Jheng-Hong Yang, Rodrigo Nogueira, Ming-Feng Tsai, Chuan-Ju Wang, and Jimmy Lin. 2020. Query reformulation using query history for passage retrieval in conversational search. *arXiv preprint arXiv:2005.02230* (2020).

[57] Xiaoyong Liu and W Bruce Croft. 2004. Cluster-based retrieval using language models. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*. 186–193.

[58] Xiaoyong Liu and W Bruce Croft. 2008. Evaluating text representations for retrieval of the best group of documents. In *European Conference on Information Retrieval*. Springer, 454–462.

[59] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-Task Deep Neural Networks for Natural Language Understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 4487–4496.

[60] Sean MacAvaney, Andrew Yates, Kai Hui, and Ophir Frieder. 2019. Content-based weak supervision for ad-hoc re-ranking. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 993–996.

[61] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems* 26 (2013).

[62] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *CoCo@ NIPs*.

[63] Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A New Benchmark for Natural Language Understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 4885–4901.

[64] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).

[65] Adi Omari, David Carmel, Oleg Rokhlenko, and Idan Szpektor. 2016. Novelty based ranking of human answers for community questions. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 215–224.

[66] Harshith Padigela, Hamed Zamani, and W Bruce Croft. 2019. Investigating the successes and failures of BERT for passage re-ranking. *arXiv preprint arXiv:1905.01758* (2019).

[67] Ellie Pavlick and Chris Callison-Burch. 2016. Simple PPDB: A paraphrase database for simplification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 143–148.

[68] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*. 1532–1543.

[69] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365* (2018).

[70] Jason Phang, Thibault Févry, and Samuel R Bowman. 2018. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv preprint arXiv:1811.01088* (2018).

[71] Chen Qu, Liu Yang, Cen Chen, W Bruce Croft, Kalpesh Krishna, and Mohit Iyyer. 2021. Weakly-supervised open-retrieval conversational question answering. In *European Conference on Information Retrieval*. Springer, 529–543.

[72] Chen Qu, Liu Yang, Cen Chen, Minghui Qiu, W Bruce Croft, and Mohit Iyyer. 2020. Open-retrieval conversational question answering. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 539–548.

[73] Chen Qu, Liu Yang, W Bruce Croft, Yongfeng Zhang, Johanne R Trippas, and Minghui Qiu. 2019. User intent prediction in information-seeking conversations. In *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval*. 25–33.

[74] Chen Qu, Liu Yang, Minghui Qiu, W Bruce Croft, Yongfeng Zhang, and Mohit Iyyer. 2019. BERT with history answer embedding for conversational question answering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1133–1136.

[75] Chen Qu, Liu Yang, Minghui Qiu, Yongfeng Zhang, Cen Chen, W Bruce Croft, and Mohit Iyyer. 2019. Attentive history selection for conversational question answering. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1391–1400.

[76] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

[77] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* 21, 140 (2020), 1–67.

[78] Fiana Raiber and Oren Kurland. 2013. Ranking document clusters using markov random fields. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. 333–342.

[79] Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 784–789.

[80] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 2383–2392.

[81] Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. 2016. Data programming: Creating large training sets, quickly. *Advances in neural information processing systems* 29 (2016).

[82] Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics* 7 (2019), 249–266.

[83] Rodrygo LT Santos, Craig Macdonald, and Iadh Ounis. 2010. Exploiting query reformulations for web search result diversification. In *Proceedings of the 19th international conference on World wide web*. 881–890.

[84] Sheikh Muhammad Sarwar, Raghavendra Addanki, Ali Montazeralghaem, Soumyabrata Pal, and James Allan. 2020. Search Result Diversification with Guarantee of Topic Proportionality. In *Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval*. 53–60.

[85] Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1073–1083.

[86] Eilon Sheetrit and Oren Kurland. 2019. Cluster-based focused retrieval. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2305–2308.

[87] Eilon Sheetrit, Anna Shtok, Oren Kurland, and Igal Shprincis. 2018. Testing the cluster hypothesis with focused and graded relevance judgments. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1173–1176.

[88] Ian M Soboroff, Nick Craswell, Charles L Clarke, and Gordon Cormack. 2011. *Overview of the trec 2011 web track*. Technical Report.

[89] Amir Soleimani, Christof Monz, and Marcel Worring. 2021. NLQuAD: A Non-Factoid Long Question Answering Data Set. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. 1245–1255.

[90] Anastasios Tombros, Robert Villa, and Cornelis J Van Rijsbergen. 2002. The effectiveness of query-specific hierarchic clustering in information retrieval. *Information processing & management* 38, 4 (2002), 559–582.

[91] Lakshmi Vikraman, W Bruce Croft, and Brendan O'Connor. 2018. Exploring diversification in non-factoid question answering. In *Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval*. 223–226.

[92] Lakshmi Vikraman, Ali Montazeralghaem, Helia Hashemi, W Bruce Croft, and James Allan. 2021. Passage Similarity and Diversification in Non-factoid Question Answering. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*. 271–280.

[93] Nikos Voskarides, Dan Li, Pengjie Ren, Evangelos Kanoulas, and Maarten de Rijke. 2020. Query resolution for conversational search with limited supervision. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 921–930.

[94] Di Wang and Eric Nyberg. 2015. A long short-term memory model for answer sentence selection in question answering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. 707–712.

[95] Shuai Wang, Shengyao Zhuang, and Guido Zuccon. 2021. Bert-based dense retrievers require interpolation with bm25 for effective passage retrieval. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*. 317–324.

[96] Xing Wei and W Bruce Croft. 2006. LDA-based document models for ad-hoc retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 178–185.

[97] Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, 1112–1122.

[98] Long Xia, Jun Xu, Yanyan Lan, Jiafeng Guo, and Xueqi Cheng. 2015. Learning maximal marginal relevance model via directly optimizing diversity evaluation measures. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. 113–122.

[99] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *International Conference on Learning Representations*.

[100] Peng Xu, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. 2019. Passage Ranking with Weak Supervsion. *arXiv preprint arXiv:1905.05910* (2019).

[101] Liu Yang, Qingyao Ai, Damiano Spina, Ruey-Cheng Chen, Liang Pang, W Bruce Croft, Jiafeng Guo, and Falk Scholer. 2016. Beyond factoid QA: effective methods for non-factoid answer sentence retrieval. In *European Conference on Information Retrieval*. Springer, 115–128.

[102] Liu Yang, Minghui Qiu, Chen Qu, Cen Chen, Jiafeng Guo, Yongfeng Zhang, W Bruce Croft, and Haiqing Chen. 2020. IART: Intent-aware Response Ranking with Transformers in Information-seeking Conversation Systems. In *Proceedings of The Web Conference 2020*. 2592–2598.

[103] Liu Yang, Minghui Qiu, Chen Qu, Jiafeng Guo, Yongfeng Zhang, W Bruce Croft, Jun Huang, and Haiqing Chen. 2018. Response ranking with deep matching networks and external knowledge in information-seeking conversation systems. In *The 41st international acm sigir conference on research & development in information retrieval*. 245–254.

[104] Shi Yu, Jiahua Liu, Jingqin Yang, Chenyan Xiong, Paul Bennett, Jianfeng Gao, and Zhiyuan Liu. 2020. Few-shot generative conversational query rewriting. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 1933–1936.

[105] Shi Yu, Zhenghao Liu, Chenyan Xiong, Tao Feng, and Zhiyuan Liu. 2021. Few-shot conversational dense retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 829–838.

[106] Yisong Yue and Thorsten Joachims. 2008. Predicting diverse subsets using structural SVMs. In *Proceedings of the 25th international conference on Machine learning*. 1224–1231.

[107] Hamed Zamani and W. Bruce Croft. 2018. On the Theory of Weak Supervision for Information Retrieval. In *Proceedings of the Fourth International Conference on the Theory of Information Retrieval (ICTIR '18)*. 147 – 154.

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

[109] Chengxiang Zhai and John Lafferty. 2017. A study of smoothing methods for language models applied to ad hoc information retrieval. In *ACM SIGIR Forum*, Vol. 51. ACM, 268–276.

[110] Lingjun Zhao, Rabih Zbib, Zhuolin Jiang, Damianos Karakos, and Zhongqiang Huang. 2019. Weakly supervised attentional model for low resource ad-hoc cross-lingual information retrieval. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*. 259–264.

[111] Yadong Zhu, Yanyan Lan, Jiafeng Guo, Xueqi Cheng, and Shuzi Niu. 2014. Learning for search result diversification. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. 293–302.