Characterizing Question Facets for Complex Answer Retrieval

Sean MacAvaney
IRLab, Georgetown University
sean@ir.cs.georgetown.edu

Andrew Yates
Max Planck Institute for Informatics
ayates@mpi-inf.mpg.de

Arman Cohan, Luca Soldaini
IRLab, Georgetown University
{arman,luca}@ir.cs.georgetown.edu

Kai Hui
SAP SE
kai.hui@sap.com

Nazli Goharian, Ophir Frieder
IRLab, Georgetown University
{nazli,ophir}@ir.cs.georgetown.edu

ABSTRACT

Complex answer retrieval (CAR) is the process of retrieving answers to questions that have multifaceted or nuanced answers. In this work, we present two novel approaches for CAR based on the observation that question facets can vary in utility: from structural (facets that can apply to many similar topics, such as 'History') to topical (facets that are specific to the question's topic, such as the 'Westward expansion' of the United States). We first explore a way to incorporate facet utility into ranking models during query term score combination. We then explore a general approach to reform the structure of ranking models to aid in learning of facet utility in the query-document term matching phase. When we use our techniques with a leading neural ranker on the TREC CAR dataset, our methods rank first in the 2017 TREC CAR benchmark, and yield up to 26% higher performance than the next best method.

1 INTRODUCTION

As people become more comfortable using question answering systems, it is inevitable that they will begin to expect the systems to answer questions with complex answers. For instance, even the seemingly simple question *"Is cheese healthy?"* cannot be answered with a simple 'yes' or 'no'. To fully answer the question, positive and negative qualities should be discussed, along with the strength of evidence, and conditions under which the qualities apply—a complex answer. Complex Answer Retrieval (CAR) frames this problem as an information retrieval (IR) task [2]. Given a query that consists of a topic (e.g., 'cheese'), and facets of the topic (e.g., 'health effects'), a CAR system should be able to retrieve information from a variety of sources to throughly answer the corresponding question.

CAR has similarities with existing, yet distinct, areas of research in IR. Although CAR involves passage retrieval, it is distinguishable from passage retrieval because CAR compiles multiple passages together to form complete answers. It is also different than factoid question answering (questions with a simple answer, e.g. *"Who wrote Hamlet?"*), and complex question answering (questions that themselves require reasoning, e.g. *"Which female characters are in the same room as Homer in Act III Scene I?"*).

We observe that question facets can be structural or topical. Structural facets refer to general categories of information that could apply to other entities of the same type, such as the 'History' or 'Economy' of a country. Topical facets refer to categories of information that are specific to the entity mentioned in the question, such as the 'Westward expansion' or 'Banking crisis' of the United States. (Although either facet could be asked about other topics, they are much more specific to details of the topic than structural headings.) We call this distinction *facet utility*, and explain it in detail in Section 2, along with additional background and related work. We then present and evaluate two novel approaches to CAR based on this observation and the hypothesis that it will affect how terms are matched. The first approach integrates predictors of a facet's utility into the score combination component of an answer ranker. The second approach is a technique to help any model learn to make the distinction itself by treating different facets independently. To predict facet utility, we use the heading structure of CAR queries (described in Section 2) and corpus statistics. We show how our approaches can be integrated with recent neural ranking models, and evaluate on the TREC CAR dataset. Our approaches yield favorable results compared to other known methods, achieving the top results overall and up to a 26% gain over the next best method.

2 BACKGROUND AND RELATED WORK

The first major work done with CAR frames the task in terms of Wikipedia content generation [1]. CAR fits naturally with this domain because CAR query topics and facets often correspond well with article titles and headings, respectively. Furthermore, Wikipedia itself provides an extensive source of sample queries (paths in the heading hierarchy from the title), partial answers (i.e., paragraphs), and automatic relevance judgments (paragraphs can be assumed relevant to the headings they are under). For simplicity, we use Wikipedia-focused terminology in the remainder of this work. A *heading* refers to any component of a query, and corresponds to a question topic or facet. The *title* is the first query component (topic), the *main heading* is the last component, and *intermediate heading* are any headings between the two (if any). The main and intermediate headings represent the facet of interest to the topic. Example queries using this terminology are given in Table 1.

A central challenge of CAR is resolving *facet utility*. Due to the structure of CAR queries as a list of headings, we generalize the concept to *heading utility*—the idea that headings (i.e., question topics and facets) can serve a variety of functions in an article.

SIGIR '18, July 8–12, 2018, Ann Arbor, MI, USA
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5657-2/18/07…$15.00
https://doi.org/10.1145/3209978.3210135

Table 1: Example CAR queries from Wikipedia by heading position. Some queries have no intermediate headings.

Title		Intermediate Heading(s)		Main Heading
Cheese	»	*(none)*	»	Nutrition and health
Green sea turtle	»	Ecology and behavior	»	Life cycle
History of the United States	»	20th Century	»	Imperialism
Disturbance (ecology)	»	*(none)*	»	Cyclic disturbance
Medical tourism	»	Destinations » Europe	»	Finland

Table 2: Example contextual vectors for the query "*green sea turtle » ecology and behavior » life cycle*".

	green	sea	turtle	ecology	and	behavior	life	cycle
position_title	1	1	1	0	0	0	0	0
position_inter	0	0	0	1	1	1	0	0
position_main	0	0	0	0	0	0	1	1
heading_frequency	0	0	0	3	3	3	3	3

We distinguish between structural and topical headings. We define *structural headings* as headings that serve a structural purpose for an article—general question facets that could be asked about many similar topics. In contrast, *topical headings* describe details that are specific to the particular topic. For instance, *"cooking and eating"* is a structural heading for Cheese (one would expect it to be found in other food-related articles), whereas *"cheeseboard"* is a topical heading because it relates specifically to the topic of the article. Because the terminology in structural headings is necessarily more generic (they accommodate many topics), we predict that terms found in these headings are less likely to appear verbatim in relevant paragraphs than terms in topical headings. Thus, modeling this behavior should improve performance on CAR because it will be able to learn which terms are less important. Previous work does not model facet utility, treating all headings equally by concatenating their components.

Nanni et al. [8] presents a survey of prominent general domain ranking and query expansion approaches for CAR. They test one deep neural model (Duet [7]), and find that it outperforms the other approaches, including BM25, cosine similarity with TF-IDF and word embeddings, and a learning-to-rank approach. The recent 2017 TREC track focused on CAR [2]. This track yielded both manual relevance judgments for evaluation of CAR systems, and a variety of new CAR approaches (seven teams participated). One prominent approach used a sequential dependence model [5]. They modified the approach for CAR by limiting ordered ngrams to those found within a single heading, and unordered ngrams to only inter-heading pairs. Another approach uses a Siamese attention network [6], including topic features extracted from DBPedia. While this approach does distinguish the title from other headings, it only uses it for query expansion and related entity extraction. Another submission applied a reinforcement learning-based query reformulation approach to CAR [9].

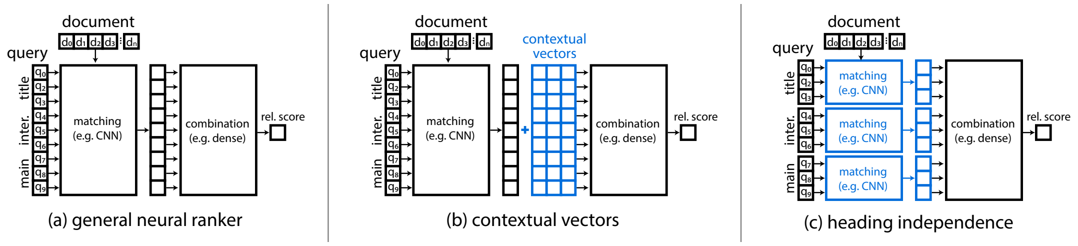
3 METHOD

Since previous work shows that neural-based rankers have potential for CAR, we focus on an approach that can be adapted for various neural rankers. Many leading interaction-focused neural rankers share a similar two-phase architecture, as shown in Figure 1a. Phase 1 performs matching of query terms to document terms, and phase 2 combines the matching results to produce a final relevance score. For instance, DRMM [3] uses a feed-forward histogram matching network, and a term gating combination network to predict relevance. MatchPyramid [10] uses hierarchal convolution for matching, followed by a dense layer for aggregation. Similarly, PACRR [4] uses a max-pooled convolution phase for matching, and a recurrent or dense combination phase. Finally, DeepRank [11] generates query contexts and uses a convolutional layer to generate local relevance representations as a matching phase, and uses a term gating mechanism for combination. We present two approaches to model facet utility by modifying this generalized neural ranking structure. The first approach applies *contextual vectors* in the combination phase (Figure 1b), and the second approach splits the input into independent matching phases (Figure 1c).

Contextual vectors. In the combination phase, signals across query terms are combined to produce a relevance score, so it is natural to include information here to provide additional context about each query term when combining the results. For instance, PACRR includes the inverse document frequency (IDF) in its combination layer, allowing the model to learn how to weight results based on this statistic [4]. We use this phase to inform the model about heading utility based on predictions about the distinction between structural and topical headings. We call these *contextual vectors*, since they provide context in the CAR domain. The intuition is that by providing the model with estimators of heading utility, the model will learn which terms to weight higher. Here we explore two types of contextual vectors: *heading position* (HP) and *heading frequency* (HF).

When distinguishing between structural and topical headings, it is important to consider the position itself in the query. For instance, since the title is the question topic, it is necessarily topical. Furthermore, it is reasonable to suspect that intermediate headings will often be structural because they assist in the organization of an article. Main headings may either be structural or topical, depending on the question itself. Thus, for *heading position* contextual vectors, we use a simple indicator to distinguish whether a term is from the title, an intermediate, or the main heading. An example is given in Figure 2.

Another approach to modeling structural and topical headings using contextual vectors is to examine the prevalence of a given heading. This is based on the intuition that structural headings should appear in many similar documents, whereas the usage of topical headings should be less widespread. For instance, the structural heading *"Nutrition and health"* in the article *Cheese* also appears in articles entitled *Beef*, *Raisin*, *Miso*, and others, whereas the topical *"Cheeseboard"* heading only also appears as the title of a disambiguation page. We model this behavior using *heading usage frequency*: $frq(h) = \frac{\sum_{a \in C} I(h \in a)}{|C|}$. That is, the probability that a given article a in corpus C contains heading h, given the indicator function I. Heading usage frequencies very close to 0 include titles and other content-specific headings like *Cheeseboard*. Due to the wide variety of Wikipedia articles, most probabilities are very low. Therefore, we stratify the scores by percentile, grouping similarly-common headings together. Based on pilot studies, we found the (1) 60th, (2) 90th, and (3) 99th percentiles to be effective breakpoints. We use



(a) general neural ranker

(b) contextual vectors

(c) heading independence

Figure 1: (a) General interaction-focused ranking architecture, with matching and combination phases (unmodified). (b) Modified architecture, including contextual vectors for combination. (c) Modified architecture, splitting for heading independence.

complete, case insensitive heading matches. Unknown headings are assumed to be infrequent, and belong to the 0th percentile. An example of this vector is given in Table 2.

Heading independence. Since contextual vectors are applied in the combination phase, they have no effect on the criteria constituting a strong signal from the matching phase. However, we hypothesize that facet utility can also be important when matching. For instance, a structural heading like *"History"* might have a lower matching threshold, allowing matches of similar words terms such as *"early"* or *"was"* (both of which have a lower WORD2VEC cosine similarity score to *"history"* than functionally-related word pairs, such as *"cheese"* and *"chocolate"*).

Thus, we propose a method called *heading independence*. With this approach, we modify the structure of a generic neural IR model by splitting the matching stage into three independent parts: one for the title, one for intermediate headings, and one for the main heading. Each sub-matching phase operates independently as it otherwise would for the combined query. Then, the results are combined using the same combination logic of the original model (e.g., a dense or recurrent layer). This allows the model to learn separated logic for different heading components. The reasoning behind the split by query component is the same as the reasoning behind using heading position vectors: the title is topical, whereas intermediate headings are likely structural, and the main heading could be either. With separate matching logic for each, the model should be able to more easily distinguish between the types.

An added benefit of this approach is that it improves heading alignment in the combination phase. When headings are simply concatenated (even with a symbol to indicate a change in headings), the alignment of each query component will vary among queries. Since the output of each matching stage is fixed in size, the locations of each query component will be consistent among queries. We suspect that this is particularly useful when using dense combination.

4 EXPERIMENTAL SETUP

Dataset. TREC CAR provides several sets of queries based on a recent dump of Wikipedia [1, 2]. Queries in each set are generated from the heading structure of an article, where each query represents a path from the article title down to the main heading. Each query also includes automatic relevance judgments based on the assumption that paragraphs under a given heading are relevant to the query with that main heading. Half of the dump belongs to the `train` set, which is split into 5 folds. We use folds 1 and 2 in this work, consisting of $873,746$ queries and $2.2M$ automatic relevance judgments (more data than this was not required for our models to converge). The `test200` set contains 1,860 queries and $4.7k$ automatic relevance judgments. The `benchmarkY1test` set contains 2,125 queries and $5.8k$ automatic relevance judgments. It also includes $30k$ manual relevance judgments, ranging from *Trash* (-2) to *Must be mentioned* (3). The `paragraphcorpus` is a collection of $30M$ paragraphs from the Wikipedia dump with no article or heading structure provided, functioning as a source of answers for retrieval.

Model integration. We evaluate our contextual vector and heading independence approaches using the Position-Aware Convolutional Recurrent Relevance neural IR architecture (PACRR) [4], which is a strong neural retrieval model with a structure that naturally lends itself to incorporating contextual vectors and heading independence signals. We refer the reader to Hui et al. [4] for full details about the model, but we give a short description here to provide details about how our approach is integrated. PACRR first processes square convolutional filters over a $q \times d$ query-document similarity matrix, where each cell represents similarity scores between the corresponding query and document term. The filters are max-pooled for each cell, and the scores are k-max pooled over each query term ($k = 2$). Then a dense layer combines the scores (along with term IDF scores) to yield a final relevance score for the query-document pair. For runs that include *contextual vectors*, we append them to each term (alongside IDF) during combination. For *heading independence*, we use separate convolution and pooling layers, followed by a dense layer for each heading component. We also explore using the heading frequency contextual vector when using heading independence (included after the pooling layer), and before the independent dense layer.

Training and evaluation. We train the models on samples from `train.fold1` and `train.fold2`. Positive training examples come from the automatic relevance judgments, whereas negative training examples are selected from the top non-relevant BM25 results for the given query. Each model is trained for 80 iterations, and the top training iteration is selected using the R-Prec on `test200`. Evaluation is conducted with automatic and manual judgments on `benchmarkY1test`. The results are based on an initial ranking of the top 100 BM25 results for each query. We report Mean Average Precision (MAP), R-Precision (R-Prec), Mean Reciprocal Rank (MRR), and normalized Discounted Cumulative Gain (nDCG) of each variation (all four official TREC CAR metrics).

Table 3: **Performance results on *benchmarkY1test*. The top value is in bold. Records marked with * are based on official TREC runs, and had top results included in the manual assessment pool. Significant results compared to the unmodified PACRR model are marked with ▲ and ▼ (paired t-test, 95% confidence). The abbreviations for our methods are as follows: HP is the heading position contextual vector; HF is the heading frequency contextual vector; HI is heading independence.**

	Automatic				Manual			
Approach	MAP	R-Prec	MRR	nDCG	MAP	R-Prec	MRR	nDCG
PACRR (no modification)	0.164	0.131	0.247	0.254	0.208	0.219	0.445	0.403
PACRR + HP*	▲ 0.170	0.135	▲ 0.258	▲ 0.260	0.209	0.218	0.452	0.406
PACRR + HP + HF*	▲ 0.170	0.134	▲ 0.255	▲ 0.259	**▲ 0.211**	**0.221**	**0.453**	**▲ 0.408**
PACRR + HI	▲ 0.171	0.139	▲ 0.256	▲ 0.260	0.205	0.213	0.442	0.403
PACRR + HI + HF	**▲ 0.176**	**▲ 0.146**	**▲ 0.263**	**▲ 0.265**	0.204	0.214	0.440	0.401
Sequential dependence model* [5]	▼ 0.150	▼ 0.116	▼ 0.226	▼ 0.238	▼ 0.172	▼ 0.186	▼ 0.393	▼ 0.350
Siamese attention network* [6]	▼ 0.121	▼ 0.096	▼ 0.185	▼ 0.175	▼ 0.137	▼ 0.171	▼ 0.345	▼ 0.274
BM25 baseline*	▼ 0.122	▼ 0.097	▼ 0.183	▼ 0.196	▼ 0.138	▼ 0.158	▼ 0.317	▼ 0.296

5 RESULTS

We present system performance in Table 3. Our methods are compared to the unmodified PACRR model, two other top submissions to TREC CAR 2017 (sequential dependency model [5] and the Siamese attention network [6]), and a BM25 baseline (which produces the initial result set that our methods re-rank).

Our method outperforms the other TREC submissions and the BM25 baseline by all metrics for both manual and automatic relevance judgments (paired t-test, 95% confidence). The method that uses heading independence (HI) and the heading frequency vector (HF) yields up to a 26% improvement over the next best approach (SDM).

Our approach also consistently outperforms the unmodified version of PACRR when evaluating using automatic relevance judgments, performing up to 11% better than the unmodified version of PACRR. Our approach occasionally does better than unmodified PACRR when evaluating with manual relevance judgments. Specifically, our approach that uses the heading position (HP) and heading frequency (HF) contextual vectors does the best overall. We acknowledge that this method (and the version with only heading position) were included as official TREC runs, yielding an advantage in the manual comparison.

This work is based on the distinction between structural and topical headings, and the differences in how they interact with terms in relevant documents. While there is no absolute distinction between the two, we presented various approaches to approximate the distinction. By plotting the *term occurrence rate* (that is, the probability that any term occurs in a relevant paragraph) for title, intermediate, and main headings, we see clear differences in the distribution (Figure 2). Particularly, the plot shows that main headings are much more likely to appear in relevant documents than title and intermediate headings. Furthermore, the distributions of intermediate and title headings are roughly opposite each other, with titles (topical) more likely to occur than intermediate headings (structural).

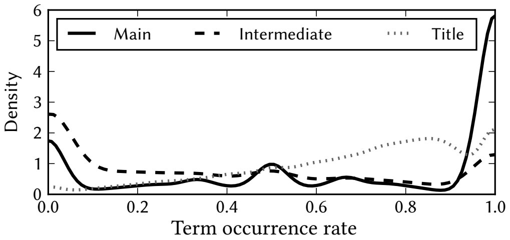


Figure 2: Kernel density estimation for main (solid), intermediate (dashed), and title (dotted) heading term occurrence rates, based on automatic judgments in train.fold0.

6 CONCLUSION

In this work, we presented an approach to the new and challenging task of complex answer retrieval. Our approach characterizes question facets by modifying a generic neural IR architecture. We explored both approaches that focus on matching (*heading independence*), and score combination (*contextual vectors*). When evaluating on the TREC CAR dataset, we achieve the top results—up to a 26% improvement over the next best method. Furthermore, our approach significantly outperforms a leading neural IR model when evaluating with both automatic and manual judgments.

REFERENCES

[1] Laura Dietz and Ben Gamari. 2017. TREC CAR: A Data Set for Complex Answer Retrieval (Version 1.5). (2017). http://trec-car.cs.unh.edu

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

[3] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *CIKM 2016*.

[4] Kai Hui, Andrew Yates, Klaus Berberich, and Gerard de Melo. 2017. PACRR: A Position-Aware Deep Model for Relevance Matching in Information Retrieval. In *EMNLP 2017*.

[5] Xinshi Lin and Wai Lam. 2017. CUIS Team for TREC 2017 CAR Track. In *TREC 2017*.

[6] Ramon Maldonado, Stuart Taylor, and Sanda M. Harabagiu. 2017. UTD HLTRI at TREC 2017: Complex Answer Retrieval Track. In *TREC 2017*.

[7] Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2017. Learning to Match Using Local and Distributed Representations of Text for Web Search. In *WWW 2017*.

[8] Federico Nanni, Bhaskar Mitra, Matt Magnusson, and Laura Dietz. 2017. Benchmark for Complex Answer Retrieval. In *ICTIR 2017*.

[9] Rodrigo Nogueira, Kyunghyun Cho, Urjitkumar Patel, and Vincent Chabot. 2017. New York University Submission to TREC-CAR 2017. In *TREC 2017*.

[10] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, and Xueqi Cheng. 2016. A Study of MatchPyramid Models on Ad-hoc Retrieval. In *NeuIR 2016*.

[11] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Jingfang Xu, and Xueqi Cheng. 2017. DeepRank: A New Deep Architecture for Relevance Ranking in Information Retrieval. In *CIKM 2017*.