Contextualized PACRR for Complex Answer Retrieval

Sean MacAvaney[1]*, Andrew Yates[2], and Kai Hui[2]

[1] Information Retrieval Lab
Georgetown University
Washington DC, USA
sean@ir.cs.georgetown.edu
[2] Max Planck Institute for Informatics
Saarbrücken, Germany
{ayates,khui}@mpi-inf.mpg.de

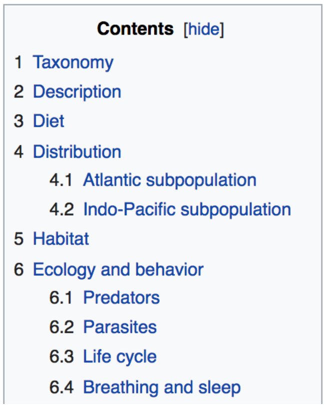
Abstract. In this work, we present our submission to the TREC Complex Answer Retrieval (CAR) task. Our approach uses a variation of the Position-Aware Convolutional Recurrent Relevance Matching (PACRR) [2] deep neural model to re-rank passages. Modifications include an expanded convolutional kernel size, and contextual vectors to capture heading type (e.g. title), heading frequency, and query term occurrence frequency. We submitted three runs for human relevance judgments by TREC varying which contextual vectors are included, and the number of negative samples used when training. Our approach yields a MAP of 0.241, R-Prec of 0.321, and MRR of 0.520 when using the term occurrence frequency run in the lenient evaluation environment.

1 Introduction

Most existing research in question answering has focused on the retrieval of answers to questions that have a single answer. Complex Answer Retrieval (CAR) aims to answer questions that have varied, multi-faceted answers. For instance, somebody who searches for “What is the life cycle of a green sea turtle?” probably expects several paragraphs that describing the birth, maturity, migration, and death of the creature. A system that could pull the information for each facet from the best source would results in a complete answer to the user’s question, and would save the user the time of searching through these sources themselves. The benefits are even more clear when controversial questions are asked. For instance, a question like “What effect will the new health care plan have on my premiums?” would ideally pull a variety of opinions from reputable sources, and let the user make judgments about the merits of each take.

The TREC CAR task aims to encourage research into this area by providing benchmarks and a common dataset. The task uses Wikipedia as both a collection

[*] This work was conducted at an internship at the Max Planck Institute for Informatics.



Contents [hide]

1 Taxonomy
2 Description
3 Diet
4 Distribution
4.1 Atlantic subpopulation
4.2 Indo-Pacific subpopulation
5 Habitat
6 Ecology and behavior
6.1 Predators
6.2 Parasites
6.3 Life cycle
6.4 Breathing and sleep

Fig. 1. Example headings found in the Wikipedia article 'Green sea turtle'

of documents and a source of queries. Headings within an article are used as queries. For instance, the article titled "Green Sea Turtle" with the headings in Figure 1 has queries of "Green Sea Turtle » Taxonomy", "Green Sea Turtle » Ecology and behavior » Life cycle", etc. Each paragraph in Wikipedia is treated as an independent document; they must be treated independently of their context to avoid overfitting approaches to Wikipedia itself. There is an implicit relevance between each heading and the paragraphs that appear under it, which serves as an enormous (albeit weak) source of training data. TREC CAR collects human relevance judgments for system comparisons.

Throughout the paper we use the following terminology. The *heading chain* refers to a single path in the *heading hierarchy* of a document. The task treats heading chains as queries. For simplicity, we refer to every element within the heading hierarchy as a *heading*, even though the first is the article title. *Title* is reserved for the first heading, and *main heading* describes the last heading in a given chain. Any headings between them are called *intermediate headings*. A heading chain will always have a title and main heading, but does not necessarily have any intermediate headings. Example queries are given in Table 1.

Table 1. Example queries

Query	Title	Intermediate Heading(s)	Main Heading
q_{turtle}	"Green Sea Turtle"	["Ecology and behavior"]	"Life cycle"
$q_{us-hist}$	"History of the United States"	["20th Century"]	"Imperialism"
$q_{disturb}$	"Disturbance (ecology)"	[]	"Cyclic disturbance"
$q_{medical}$	"Medical tourism"	["Destinations", "Europe"]	"Finland"

We identify that a central challenge of the CAR task is addressing *discourse context*. Individual paragraphs of well–written articles will often omit what could otherwise be considered critical text. For instance, only the first of the four relevant paragraphs for q_{turtle} use the phrase “Green sea turtle” in its entirety, with subsequent paragraphs simply using the head noun “turtle”. Other examples use pronouns, or are otherwise able to express the ideas with no matching terms. Since the coreferent term appears in separate paragraphs and it is necessary to treat paragraphs in isolation, coreference resolution cannot be used to work around this problem. Headings can also introduce context themselves. For instance, relevant paragraphs for “History” are unlikely to use the word, but instead simply present historical events.

Related to the problem of discourse context is the problem of *heading utility*. We observe that headings can serve a variety of functions in an article. Intuition suggests that titles should appear in relevant paragraphs, but this breaks down due to discourse context. Furthermore, some titles will never appear in the text verbatim (aside from, perhaps, the lead paragraph) because they include structure. For instance, $q_{us-hist}$ and q_{eco} provide specifiers from what would otherwise be the overly-general or ambiguous topics of “History” and “Disturbance”, respectively.

The remainder of this paper is organized as follows. We first present related work and existing benchmarks. We then motivate and describe our methodology. Finally, we present and discuss our results.

2 Related work

Although Complex Answer Retrieval is a new task in TREC 2017, some work has already conducted on the CAR dataset. Some researchers presented a survey of prominent ranking and query expansion approaches, with the intent for the results to serve as a baseline for CAR participants [4]. They found that the deep neural model Duet [3] outperformed other approaches, including Okapi BM25, cosine similarity with TF-IDF and word embeddings, and a learning-to-rank approach. They also propose a ‘safe environment’ for training in which the set of candidate paragraphs is limited, allowing for faster experimentation. Others have successfully used the CAR dataset to evaluate a neural reinforcement query reformulation approach [5]. They found that recurrent neural networks marginally outperformed a version of their approach that used convolutional neural networks.

Recently, improvements have been observed for ad-hoc information retrieval using neural methods. DRMM uses term occurrence histograms and a dense deep neural architecture to predict document relevance [1]. Duet jointly models local and global document interactions using convolutional neural networks [3]. MatchPyramid uses a query-document similarity matrix and a convolutional hierarchy [6]. Most recently, the PACRR architecture was introduced, which uses convolutional filters and k-max pooling to achieve state-of-the-art ad-hoc neural retrieval results [2].

3 Method

Due to the observed effectiveness of neural IR reranking techniques [1,6,3,2] and the positive preliminary results for complex answer retrieval using neural IR techniques [4], we present a technique technique that builds upon a state-of-the-art neural information retrieval architecture (namely, PACRR [2]). By operating on word embedding similarity scores instead of exact matches, the PACRR architecture partially addresses some discourse context problems. For example, both "green" and "sea" have a similarity score of 0.48 to "turtle", providing a weak signal in lieu of no signal. This, combined with partial n-gram matches, means that architecture is relatively effective at the CAR task with no modifications. Nevertheless, we present additional considerations and structural changes that are aimed to address the discourse context problem, the heading utility problem, and the CAR task in general.

3.1 Query formulation

The trivial approach to formulating a query based on the heading chain is to simply concatenate all the headings. This, however, can lead to some very long queries. (The longest query observed in training data using this technique was 54 tokens long.) For some approaches, such as BM25, this is inconsequential. However, for neural IR approaches that process fixed-size query-document similarity matrices, this causes problems. Either the matrix size needs to be large enough to handle the longest query, or the query need to be reduced. Since such a large matrix size would contribute considerably to processing time, the only practical approach is to trim terms off the query.

Previous work has either taken the approach of truncating the query to a fixed size l or removing the terms with the lowest inverse document frequency (IDF), often being stopwords [2]. Neither approach is ideal here. Truncating can leave off critical details about the query. For instance, removing "Life cycle" from q_{turtle} would over-generalize the query. Concatenating the headings in a different order could not be done in a reliable way because of the heading utility problem: it is unclear which headings are most important for a given query.

The approach we take is to remove terms by IDF. Terms in intermediate headings are removed before terms from the title and from the main heading. This is based on the intuition that the context that the middle headings brings is less significant, e.g. in $q_{us-hist}$. It is worth noting that, although not ideal, only 0.6% of queries are affected by this procedure with a reasonable $l = 18$.

We leave further evaluation of this approach—as well as alternate approaches that challenge the concatenation assumption—to future work.

3.2 Heading position

Under the concatenation assumption, there exists nothing structural about the model to inform about the position of each token within the heading chain. We consider this a deficiency because heading utility can certainly be influenced by the position of the heading. For instance, while not always true, titles often appear in full. Thus, we include three binary vectors as additional input to PACRR's dense layer. One indicates whether a given position corresponds to a term in the title, another if it corresponds to an term in an intermediate heading, and a third that indicates when a position belongs to the main heading. An example of these vectors for q_{turtle} is given in Table 2.

Table 2. Example contextual vectors for q_{turtle}

	Green	Sea	Turtle	Ecology	and	behavior	Life	cycle
pos_title	1	1	1	0	0	0	0	0
pos_inter	0	0	0	1	1	1	0	0
pos_main	0	0	0	0	0	0	1	1
frq_strata	0	0	0	3	3	3	3	3
occ_prob	0.6	0.5	0.6	0.1	0.8	0.2	0.1	0.3
occ_prob_exp	0.6	0.5	0.7	0.1	0.8	0.3	0.9	0.3

3.3 Heading usage frequency

One hypothesis is that headings exist on a spectrum from structural to content-driven. In q_{turtle}, we consider the "Ecology and behavior" and "Life cycle" headings structural because one would expect a similar pattern in other articles about organisms. "Cyclic disturbance" in $q_{distrub}$ would be considered content-driven because it relates to a detail exclusive to the topic of ecological disturbance. We suspect that structural headings are less likely to appear verbatim in relevant paragraphs, and instead use related language. For instance, this might include terms like "birth" or "maturity" in "Life cycle", while never using the words "life" or "cycle". On the other hand, it would be difficult to adequately describe the topic of content-driven headings without using exact terminology due to the specialized language.

To approximate this spectrum, we calculate *heading usage frequency*, defined as follows:

$$frq(h) = \frac{\sum_{a \in C} I(h \in a)}{|C|}$$

That is, the probability that a given article a in corpus C contains heading h, given the indicator function I. To include these values in the model, stratify them by fixed percentile ranges. We include a vector of length l where each value corresponds to the stratum index of the corresponding heading. (In this work, we use the following percentiles (1) 60th, (2) 90th, (3) 99th. A score of 0 indicates that the heading did not meet the 60th percentile threshold.) Values close to 0 include article titles and other content-specific headings like "Cyclic disturbance". On the other end, there are headings like "Life cycle" and "History", both of which are in the 99th percentile. Complete, case insensitive heading matches are used here, so all the terms in "History of the United States" belong to stratum 0, even though "History" belongs to the 99th percentile. Unknown headings are assumed to belong to the 0th percentile. An example of this vector for q_{turtle} is given in Table 2.

3.4 Term occurrence

Term occurrence takes a local perspective on estimating heading utility. For any given term t, this feature estimates how likely it is to occur in a relevant paragraph under a heading with that term. That is,

$$occ(t) = \frac{\sum_{h \in C} \sum_{p \in rel(h)} I(t \in h \land t \in p)}{\sum_{h \in C} I(t \in h)}$$

where C is the training corpus, h is a heading, and $rel(h)$ retrieves relevant paragraphs for a given heading. Here, term matches are is determined by Porter Stemmer, with numerals folded down to a single digit placeholder. The term occurrence estimation is included in the model as another contextual vector, where each value is the *occ* value for the corresponding term. Unknown values are assumed to have a value of 0. An example is shown as occ_prob in Table 2.

Since the PACRR model "sees" values as word embedding similarity scores that allow for inexact term matches, we also an extension to term occurrence. As hypothesized with the heading usage frequency, structural headings are less likely to appear in relevant paragraphs than content-based headings. To account for these observations, we expand the definition of term matches for the purpose of the term occurrence score as follows. For any heading with a heading usage frequency greater than threshold t_{frq}, we also include matches for the top n_{exp} query-expanded terms. Query expansion is accomplished by taking the 50 terms with the highest BM25 score from all relevant documents in the training corpus. An example is shown as occ_prob_exp in Table 2. Notice how the value for "life" increases significantly when expansion is enabled. This indicates that paragraphs under "life" headings are not likely to contain the word itself, but are likely to contain words that are similar to it.

4 Evaluation

We submitted three runs to TREC CAR 2017: *nn6_pos*, *nn4_pos_hperc*, and *nn6_pos_tprob*. All three include the heading position contextual vector (*pos*). One includes the heading usage frequency vector (*hprec*), and another includes the query-expanded term occurrence vector (*tprob*). We also varied the number of negative training samples per positive sample with values of 4 (*nn4*) and 6 (*nn6*). These configurations produced the best results in the 'safe environment' described in [4]. We report our system performance in terms of Mean Average Precision (MAP), R-Precision (R-Prec), and Mean Reciprocal Rank (MRR).

TREC provided system performance results for three environments. *Automatic* uses implicit paragraph relevance, based on paragraph containment under a given heading. *Manual* judgments grade results based on the following grades: must be mentioned (3); should be mentioned (2); can be mentioned (1); roughly on topic (0); non-relevant (-1); and trash (-2). The *lenient* variation of the manual evaluation treats paragraphs that are 'roughly on topic' as positive by shifting all the grades by 1, except trash which remains at -2.

We present our results for each run and environment in Table 3. All three runs perform comparably in each environment; a paired t-test indicates that there is no statistically significant difference between any two pairs of runs within a given environment. However, we observe that nn4_pos_hperc performs marginally better than the others in the automatic and manual environments, while nn6_pos_tprob performs the best in the lenient environment by a larger margin. This could be attributed to the query expansion approach when calculating the term probabilities; the expansion includes terms that may be on topic, whereas the other approaches have no mechanism to capture terms that are roughly on topic.

Table 3. Results for each of the three runs under various evaluation environments.

Run	MAP	R-Prec	MRR
Automatic			
nn6_pos	0.144	0.111	0.216
nn4_pos_hperc	**0.148**	**0.116**	**0.224**
nn6_pos_tprob	0.140	0.107	0.213
Manual			
nn6_pos	0.197	0.209	0.417
nn4_pos_hperc	**0.201**	**0.213**	0.418
nn6_pos_tprob	0.198	0.206	**0.419**
Manual (lenient)			
nn6_pos	0.235	0.311	0.499
nn4_pos_hperc	0.234	0.311	0.505
nn6_pos_tprob	**0.241**	**0.321**	**0.520**

5 Conclusion

In this work, we presented a deep neural approach for the TREC answer retrieval task. We showed that including a measure of term occurrence frequency yields improved results.

References

1. Guo, J., Fan, Y., Ai, Q., Croft, W.B.: A deep relevance matching model for ad-hoc retrieval. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. pp. 55–64. ACM (2016)

2. Hui, K., Yates, A., Berberich, K., de Melo, G.: A position-aware deep model for relevance matching in information retrieval. arXiv preprint arXiv:1704.03940 (2017)
3. Mitra, B., Diaz, F., Craswell, N.: Learning to match using local and distributed representations of text for web search. In: Proceedings of WWW 2017. ACM (2017)
4. Nanni, F., Mitra, B., Magnusson, M., Dietz, L.: Benchmark for complex answer retrieval. CoRR abs/1705.04803 (2017), http://arxiv.org/abs/1705.04803
5. Nogueira, R., Cho, K.: Task-oriented query reformulation with reinforcement learning. In: EMNLP (2017)
6. Pang, L., Lan, Y., Guo, J., Xu, J., Cheng, X.: A study of matchpyramid models on ad-hoc retrieval. CoRR abs/1606.04648 (2016)