CUIS Team at TREC 2018 CAR Track

Xinshi Lin
The Chinese University of Hong Kong
xslin@se.cuhk.edu.hk

Wai Lam
The Chinese University of Hong Kong
wlam@se.cuhk.edu.hk

ABSTRACT

We participated in the Complex Answer Retrieval(CAR) Track at TREC 2018. We propose a Markov random field based framework concerning unigrams, bigrams and concepts from diffrent query sections. Besides, we employ a language modelling framework facilitating the Wikipedia article information and query entity mentions. Our best passage run achieves NDCG@5 of 0.3503 and MAP of 0.1715.

1 INTRODUCTION

The TREC 2018 Complex Answer Retrieval (CAR) Track provides a forum for soliciting works in meeting complex information needs with long answers. The task and related dataset are based on the assumption that each Wikipedia page represents a complex topic, with further details under each sections. The goal of the task is presented as such: given an outline of a page (in the form of the page title and hierarchical section headings), retrieve a ranking of paragraphs for each section [2]. A complex query can be long and composed of several query sections. Each complex query indicate one or several aspects of the article stub. For example, the complex query "new york yankees/rivalries" concerns the competition between baseball teams. This brings challenges to existing approaches to model term dependencies between different query sections and differentiate their importance.

We propose a Markov random field based model where unigrams, bigrams and phrases generated from different query sections. Besides, our system incorporates the Wikipedia article information and query entity mentions based on a Dirichlet prior smoothed language modeling framework.

2 MODEL DESCRIPTION

2.1 Overview

Given a complex query Q and a candidate paragraph D, we propose a MRF-based framework considering D and its corresponding Wikipedia article H. The overall ranking function $F(Q, D)$ is formulated as follows:

$$\begin{aligned} F(Q,D) = & w_{para} f_{para}(Q,D) + \\ & w_{wiki} f_{wiki}(Q,H) + \\ & w_{elr} f_{elr}(E(Q), E(D)) \end{aligned} \tag{1}$$

where $f_{para}(D,Q)$ and $f_{wiki}(Q,H)$ are ranking functions for a candidate paragraph D, its corresponding Wikipedia article H respectively. The function $E()$ returns the set of entity mentions given a segment of text. The ranking function $f_{elr}(E(Q), E(D))$ scores the paragraph D concerning the occurrences of query entity mentions $E(Q)$ in the paragraph. w_{para}, w_{wiki} and w_{elr} are parameters satisfying that $w_{para} + w_{wiki} + w_{elr} = 1$.

A Markov Random Field based Paragraph Ranking Model. Assuming that a complex query is in the form of $Q = s_1 s_2 ... s_n$ where s_i denote the i-th query section. The query section s_i contains m_i query terms $q_{i_1} q_{i_2} ... q_{i_{m_i}}$. Since the sequential dependence model cannot model the term dependencies between long distance dependence. We consider query concept expansion by sampling query terms from different query sections. Figure 1 demonstrates an example of generating query phrases from the query "Green sea turtle/conservation/threats". As shown in the figure, query terms from different query sections within a distance are considered to be relevant in context. We generate new query phrases from these relevant terms. Thus, the paragraph ranking function $f_{para}(Q, D)$ is formulated as follows:

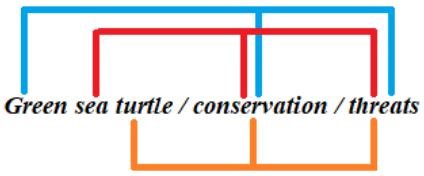

Figure 1: An example of the concept expansion scheme for the complex query "Green sea turtle/conservation/threats". Query phrases are generated by the terms connected with lines sharing the same color. The generated concepts for this query are "Green conservation", "Green threats", "conservation threats", "sea conservation", "sea threats", "turtle conservation", "turtle threats" and "conservation threats".

$$\begin{aligned} f_{para}(Q,D) = & \lambda_T \sum_{q \in Q} f_T(q,D) + \\ & \lambda_O \sum_{q_{i_x}, q_{j_y} \in Q} f_O\left(q_{i_x}, q_{j_y}, D\right) + \\ & \lambda_U \sum_{q_{i_x}, q_{j_y} \in Q} f_U\left(q_{i_x}, q_{j_y}, D\right) \end{aligned} \tag{2}$$

where i and j are the query section numbers satisfying $|i - j| \leq k$. x and y are the indices of query terms that enumerate integer in $1..m_i$ and $1..m_j$ respectively. f_T, f_O and f_U are feature functions for matching query terms, ordered bigrams and unordered bigrams within a window of 8 words. They can formulated in a uniform way as follows:

$$f_{\{T,O,U\}}(W,D) = \log \sum \frac{tf_{W,D} + \mu \frac{cf_W}{|C|}}{|D| + \mu} \tag{3}$$

where W generalizes to query terms, phrases and concepts. tf and cf denote term frequencies and collection frequencies of W respectively. μ is the Dirichlet prior.

Exploiting Wikipedia Article Information. A Wikipedia article is organized by sections. It can be represented as a tree where each

node corresponds to a section and its children represent subsections. We use the notation H to denote the tree representation of a corresponding Wikipedia article for the candidate paragraph D. We consider the occurrences of each query term q_i in each sequence of sections and find the one that has the most relevant sections to the query. The scoring function is formulated as follows according to [4]:

$$f_{wiki}(Q,H) = \max_{p \in H} \sum_{q_i \in Q} \tilde{h}_T(q_i, H, p) \tag{4}$$

$$\tilde{h}_T(q_i, H, p) = \log \frac{\sum_{s_j \in p} \beta^j \cdot tf_{q_i,s_j} + \mu_d \frac{cf_{q_i,C_d}}{|C_d|}}{\sum_{s_j \in p} \beta^j \cdot |s_j| + \mu_d} \tag{5}$$

where p enumerates each directed path in H that starts from a leaf and ends at an immediate child of the root. s_j denotes the j-th section in p. tf_{q_i,s_j} is the raw frequency of q_i in s_j. C_d is the collection of all Wikipedia articles from the knowledge sources. The smoothing parameter μ_d is set to the average document length in C_d. β is the reduction coefficient.

Exploiting Query Entity Mentions. We employ a Dirichlet prior smoothed entity language model to score the paragraph D. The ranking function f_{elr} is formulated as follows:

$$f_{elr}(E(Q), E(D)) = \sum_{e \in E(Q)} \log \frac{tf_{e,E(D)} + \mu \frac{cf_e}{|C|}}{|E(D)| + \mu} \tag{6}$$

3 EXPERIMENT

3.1 Experimental Setup

We use the TREC CAR v2.1 benchmark dataset [1] with corresponding Wikipedia dump as our knowledge base. We use Lucene to implement our model and construct the index. Each paragraph in the index is represented with three fields: *id*, *wiki article id* and *content*, which denote the paragraph identifier, the Wikipedia article the paragraph belongs to and the paragraph content. All index terms are stemmed using the Snowball stemmer and filtered with NLTK stopword list. We use Tagme [3] that indexes the Wikipedia dump on 2016-10 to annotate entity mentions in queries and paragraphs. Illegal entity mentions are removed. Our system consists of two stages. The first stage chooses top 1000 candidate passages based on Lucene's BM25 method. The second stage reranks those passages based on our proposed framework. We submitted the two passages runs and three entity runs. Each run is named after a popular vehicle in North America. The details of runs are listed as follows:

- **CUISPR:** the proposed Markov random field based model without incorporating Wikipedia article information and query entity mentions. Top 1000 candidate paragraphs for each query are reported.
- **CUIS-F150:** the proposed Markov random field based model incorporating Wikipedia article information. Top 100 candidate paragraphs for each query are reported.
- **CUIS-MX5:** the proposed Markov random field based model incorporating Wikipedia article information and query entity mentions. Top 1000 candidate paragraphs for each query are reported.
- **CUIS-Swift:** simply transform the passage ranking results in the run "CUIS-MX5". Less than top 100 candidate entities are reported for each query.
- **CUIS-XTS:** simply transform the passage ranking results in the run "CUIS-F150". Less than top 100 candidate entities are reported for each query.
- **CUIS-dogeDodge:** simply transform the passage ranking results in the run "CUIS-MX5". The results are complemented by a Dirichlet prior smoothed language model on the collection of Wikipedia articles if there are less than 100 original candidate entities for each query.

For our model, the parameters λ_T, λ_O and λ_U for SDM and FSDM are set to 0.8, 0.1 and 0.1 respectively. The parameter k is set to 4. For the Wikipedia article scoring function, the parameter β is set to 1. We train our framework on the benchmarkY1-test-public.v2.0 dataset to obtain the best values of w_{para}, w_{wiki} and w_{elr}. The Dirichlet prior μ in the ranking functions are set to the average length of paragraphs, sections or number of entities in the collection.

After we had submitted the runs to the TREC 2018, we examined our implementation and runs. We found bugs in the implementation of the scoring function f_{wiki} that results in empty output for all queries. In other words, our submitted runs were generated using the ranking system whose f_{wiki} is always zero. Here we report new runs based on the corrected ranking framework reporting top 1000 candidates. These runs are denoted with the identifier 'fix'.

3.2 Result and Discussion

Table 1: Evaluation results for passage ranking runs based on manual judgments.

run name	NDCG@5	NDCG	R-Prec	MAP
CUISPR	0.3704	0.5368	0.3230	0.3079
CUIS-F150	0.3503	0.3113	0.2094	0.1715
CUIS-MX5	0.3406	0.4218	0.1955	0.1771
CUIS-F150-fix	0.3764	0.4933	0.2512	0.2462
CUIS-MX5-fix	0.3753	0.5012	0.2499	0.2451

Table 2: Evaluation results for entity ranking runs based on automatic judgments.

run name	Automatic			
	NDCG@5	NDCG	R-Prec	MAP
CUIS-Swift	0.0635	0.0957	0.0409	0.0256
CUIS-XTS	0.0664	0.0887	0.0426	0.0264
CUIS-dogeDodge	0.0632	0.0954	0.0406	0.0254

Table 1 reports the results of our submitted passage runs and the new runs. As shown in the table, our model that only incorporates Wikipedia article information has better one-page performance. It has NDCG@5 of 0.3503 and R-Prec of 0.2094. However, the full model that incorporates both Wikipedia article information and query entity mentions outperform the previous one in terms of NDCG of 0.4218 and MAP of 0.1771. The gap between CUIS-F150

Table 3: Evaluation results for entity ranking runs based on manual judgments.

run name	Manual			
	NDCG@5	NDCG	R-Prec	MAP
CUIS-Swift	0.1654	0.2287	0.1300	0.1026
CUIS-XTS	0.1654	0.2109	0.1259	0.0988
CUIS-dogeDodge	0.1688	0.2311	0.1309	0.1042

and other variants in terms of NDCG are due to different number of top candidate paragraphs output for evaluation. Table 2 and 3 report the results of our entity runs. The submitted three runs achieve roughly equal performance. This is because most retrieved relevant paragraphs belong to the same Wikipedia article. Our proposed method cannot differentiate the importance of different Wikipedia articles.

4 CONCLUSION

We participated in the Complex Answer Retrieval (CAR) Track at TREC 2018. We propose a Markov random field based framework concerning unigrams, bigrams and phrases induced by query terms from different query sections. Besides, we employ a language modeling framework facilitating the Wikipedia article information and query entity mentions. Our best passage run achieves NDCG@5 of 0.3503 and MAP of 0.1715. Future work includes the investigation of a new framework that can differentiate the importance of query concepts and an individual entity ranking model without the help of passage ranking results.

REFERENCES

[1] Laura Dietz and Gamari Ben. 2017. Trec car: A data set for complex answer retrieval. (2017).

[2] Laura Dietz, Manisha Verma, Filip Radlinski, and Nick Craswell. 2017. TREC Complex Answer Retrieval Overview. In *Proceedings of Text REtrieval Conference (TREC)*.

[3] Faegheh Hasibi, Krisztian Balog, and Svein Erik Bratsberg. 2016. On the Reproducibility of the TAGME Entity Linking System. In *Advances in Information Retrieval*, Nicola Ferro, Fabio Crestani, Marie-Francine Moens, Josiane Mothe, Fabrizio Silvestri, Giorgio Maria Di Nunzio, Claudia Hauff, and Gianmaria Silvello (Eds.). Springer International Publishing, Cham, 436–449.

[4] Xinshi Lin, Wai Lam, and Kwun Ping Lai. 2018. Entity Retrieval in the Knowledge Graph with Hierarchical Entity Type and Content. In *Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval (ICTIR '18)*. ACM, New York, NY, USA, 211–214. https://doi.org/10.1145/3234944.3234963