Evidence Retrieval for Causal Questions Using Query Expansion And Reranking

Notebook for the Touché Lab on Argument and Causal Retrieval at CLEF 2023

Aron Gaden, Niklas Rausch, Bruno Reinhold and Lukas Zeit-Altpeter

Abstract

We present our runs to the Touché Lab on Argument and Causal Retrieval at CLEF 2023, aiming to retrieve relevant documents for given causal questions. Our approaches rely heavily on query expansions. Both a transformer-based and a technique using expansions using prior knowledge on the stated relationships are used. The retrieved documents are then retrieved using the ChatNoir search engine before being subjected to a simple reranking scheme, ranking documents higher the earlier the causal relationship in question is mentioned.

Keywords

information retrieval, query expansion, reranking, touche

1. Introduction

In this paper, we explain our approach to evidence retrieval for causal questions, as an entry for shared task 2 of Touché at CLEF 2023[1][1]. Submissions were made via TIRA[2] Evidence retrieval concerns itself with identifying whether a causal relationship between two events or actions exist. Currently, search machines do not recognize such queries, nor do they retrieve those kinds of relationships. We approached the topic under the aspects of query analysis, query expansion, and document scoring. The goal of the shared task is to retrieve and rank documents by relevance to the topic and detect the document "causal" stance, that is "for", "against", or neutral. In our case, we opt to not determine a stance, and only rank documents based on whether they make a claim about possible causal relationships. We (Team *He-Man*) propose three methods for obtaining causal relationships. The first run (*no-expansion*) is to be considered a baseline, using only the task's query as search input. The second run (*gpt-expansion*) uses responses from manual prompts to ChatGPT as expansions for each query. The third run (*causenet-expansion*) tries to find terms similar to the search terms in the *CauseNet* corpus [3]. In this paper, we explain the general methods we applied, as well as attempts we made during development. Whenever it is not obvious, we mention which run the information applies to. Finally, we discuss possible shortcomings and improvements of our approaches.

CLEF 2023: Conference and Labs of the Evaluation Forum, September 18–21, 2023, Thessaloniki, Greece

aron.gaden@uni-jena.de (A. Gaden); niklas.rausch@uni-jena.de (N. Rausch); bruno.reinhold@uni-jena.de (B. Reinhold); lukas.zeit-altpeter@uni-jena.de (L. Zeit-Altpeter)

[1] Our code is available at https://git.uni-jena.de/na93xek/he-man-causal-retrieval

2. Query Preprocessing

Finding documents that confirm or refute causal relations between two entities, forming part of the query, is a stricter constraint than just finding any documents containing those entities. Hence, we attempted to connect both cause and effect with some entity in every query. Broadly speaking, this process consists of two steps: the extraction of the two entities for which the causal relation is to be found from the tokenized queries, and what the relation between them is (i.e. which is cause and which effect). Before analyzing the terms of the query, we split each tokenized query in two parts. This was facilitated by the design of the tasks, which were all following the formula of *"Does A cause B?"* or similar, where the cause and effect would be divided by some kind of connection structure.

Sentences like *"Is there a causal relation between A and B?"* were not part of the dataset, and therefore not considered in our processing algorithm. For tokenizing the queries, we used the *NLTK* python package[2]. We extracted all the dividing patterns found in the dataset for this initial split of the query. Here, we disambiguated active and passive patterns. Active patterns (e.g. "lead to") form part of a query with the structure *"A causes B?"*, while passive patterns (e.g. "result of") reverse the relation: *"B causes A?"*. The data set with 50 questions contained 11 patterns - 7 active and 4 passive patterns. With these patterns, we could exhaustively and correctly split the queries of the data set. After splitting the query in two parts and extracting the relation between the entities, we had to identify the entities in the query parts. In an initial experiment, we queried Wikidata[3] for each n-gram contained within the partial query for every n from 1 to the length of the partial query. We assumed the largest substring of the query that returned a result from the Wikidata query as cause or effect, respectively.

We were able to identify cause and effect correctly in 33 of the 50 tasks, where the cause and effect were individually extracted correctly 41 and 44 times, respectively. Often times, this approach would leave out important context of the query term, such as *increased insurance premiums* being identified as *insurance* or *high blood pressure medication* as *high blood pressure*. Semantically, these differ drastically from the actual search term.

Hence, we used another solution with a different approach. Using Stanza [4], we built and analyzed the constituency tree of the query. With the knowledge of having two relevant search terms, we could still split the query into two parts. We assigned the longest noun phrase in each query part to be the relevant query term. Using the aforementioned rule-based strategy for identifying cause and effect, we were able to automatically recognize all cause and effect pairs in the data set correctly, hence all *he-man* runs used the latter method.

3. Query Expansion

3.1. ChatGPT Expansion

GPT stands for "Generative Pre-trained Transformer" [5]. It refers to a type of artificial neural network architecture that is widely used in natural language processing tasks, such as text generation, translation, understanding or expansion of existing text. The "generative" aspect

[2] https://www.nltk.org/

[3] https://www.wikidata.org

```
{
    "querys":
    [
        {
            "query":"Does drinking sparkling water cause weight gain
                ?",
                "expansions":[
                    {
                    "query":"Does drinking sparkling water lead to
                        weight gain?",
                    "cause":"drinking sparkling water",
                    "causal_phrase":"lead to",
                    "effect":"weight gain",
                    "voice":"active"
                 },
                ...
}
```

Figure 1: Example output of ChatGPT when prompted to rephrase a given query.

of GPT refers to the model's ability to generate new text based on the input it receives. It can produce coherent and contextually relevant text that resembles human language. However, one needs to keep in mind that ChatGPT occasionally generates inaccurate or nonsensical information. Furthermore, "Pre-trained" means that the model is trained on a large dataset before being fine-tuned for specific tasks. In the case of GPT, it is trained on a massive amount of text data, mainly from a dataset of websites called Common Crawl. This pre-training allows the model to learn patterns, grammar, vocabulary, and other linguistic features from the data. Additionally, ChatGPT has been moderated in order to make the interaction with the AI-tool more natural and safer for humans to use. "Transformer" refers to the specific architecture used in GPT. Transformers are deep learning models that rely on self-attention mechanisms [6] to process input data. They are known for their effectiveness in capturing long-range dependencies in sequences, making them suitable for tasks involving natural language.

The chat tool has been instructed to expand each of the 50 causal queries. All the original queries are interrogative sentences. The cause-and-effect relationship is clearly identifiable by a causal phase, such as e.g. "leads to" or "caused by". Furthermore, the queries can be posed in an active- or passive voice. The exact prompt is stated in figure 2.

The accuracy of the generated expansions is satisfactory in the way that the generative AI-tool follows the provided structure and generates correct and non-repetitive expansions. However, the amount of generated extension varies between four and seven. Moreover, the process of obtaining the query expansions was quite laborious. This was due to the limited and varying output size and the unsteady availability of the online tool at the time. Each query was expanded individually and composed together afterwards by concatting the `query` field of each expansion returned as `json`.

```
Expand the queries by following this json hierarchy. A query can
    only have one "cause" one "causal phrase" and one "effect"
    which are extracted from the query in an information retrieval
     task, in order to find documents containing the causal
    relationship. Find both expansions that are formulated in a
    passive and active voice. Try to use causal phrases like: "
    causes, results in, leads to, effect of, etc. Try to find at
    least 5 query expansions. the newly generated queries should
    be questions too.  Only respond in the specified json format!
```

Figure 2: Prompt to ChatGPT for generation of query expansions

3.2. CauseNet Expansion

A second way of expanding the queries derived from the given topics consists of using prior knowledge on causal questions, to improve the original query. For this we used the *CauseNet* dataset [3] that contains sources on causal relations between given concepts. We used the *CauseNet-Precision* variant, featuring around 200,000 relations between 80,000 different concepts along with snippets and IDs of pages in the *ClueWeb12* corpus, stating the supposed causal links.

We faced that the challenge that not all cause-effect-pairs had exact matches within the dataset. To find matching pairs, we embedded all *CauseNet*-concepts alongside our extracted terms from the topics using the *BERT-Uncased* model [7] and performed the matching by choosing those *CauseNet*-pairs that had the lowest mean cosine-distance to the given cause-effect-pair generated from the topics. The fulltext for the sources given for the matched relation were then retrieved using the *ChatNoir*-Cache [8]. We extracted the top 5 terms across all sources. This was done by first tokenizing the relevant documents using the pre-trained *Punkt* tokenizer provided by the *NLTK* python package[4]. The tokens were then converted to lowercase letters before counting occurences across all documents.

These terms were then added to the original topic to form the expanded query.

The topic `Is child labor an effect of poverty?` was thus expanded with the terms `child labour children education work` [5], highlighting the role of education in prevention child labour. The complete query created by query expansion is thus: `Is child labor an effect of poverty? child labour children education work`.

4. Reranking

Across all three runs one query per topic was submitted to the *ChatNoir* search engine [8]. The top 50 document ids were retrieved alongside their full text before being subjected to a custom reranking scheme. For each sentence in the retrieved document, we then detected if a postulation of the given causal relation was present. The method used here was analogous

[4] https://www.nltk.org/api/nltk.tokenize.punkt.html

[5] The duplicate of `child labour` in both original topic and expansion terms is due to the word-based tokenization that was used when extracting the sources' top words.

Table 1
All runs submitted for the retrieval task ranked by normalized discounted cumulative gain.

Team	Run	NDCG@5	95% confidence interval
He Man	heman_no_expansion_rerank	0.657	[0.564, 0.740]
Puss In Boots	puss-in-boots_baseline	0.585	[0.503, 0.673]
He Man	heman_gpt_expansion_rerank	0.374	[0.284, 0.469]
He Man	heman_causenet_expansion_rerank	0.268	[0.172, 0.368]

to the method described in section 2. We then counted the number of mentions of the given relations, weighing each occurrence by its position within the document with a weight of 1 indicating the very first sentence of the document and the weight getting closer to 0 the further back the occurence is located. All weights were then summed to get the document's occurence score.

ChatNoir returns scores for each retrieved document. The original score and the occurence score were combined using the following formula:

$$score_{reranked} = score_{ChatNoir}^{1+\frac{score_{occurrence}}{2}}$$

We experimented around with different approaches for combining the scores but came to the conclusion to use the exponent-based method to make sure that our score has a sufficient impact on the ranking. 1 is added to the score to make sure that those documents that do not contain and explicit mention are still ranked highly if *ChatNoir* returns a high enough score.

5. Evaluation

In total we submitted three different runs. Our baseline run by submitting the topics' titles to *ChatNoir* before reranking as described above. The *ChatNoir* and *ChatGPT* runs additionally expanded the title as described above before retrieving and reranking the results.

Overall, our baseline run archieved the best NDCG across all submitted runs with a NDCG of 0.657 (cf. Table 1). Both expansion techniques significally lowered the retrieval performance.

6. Conclusion

We have been successful in using constituency parsing and simple rules to identify cause and effect in all queries of the data set. Nevertheless, the approach was engineered for this particular data set and will not perform to this standard in general application. Future causal retrieval models may want to improve on this procedure.

The expansion of the queries with the help of ChatGPT worked well. Although the generation of the expansions was quite manual labor intensive due to the lack of an API. With regards to the *CauseNet* based expansion it should be noted that while most of the given cause-effect-combinations were succesfully matched to a fitting counterpart, some cases failed, resulted in nonsense expansions. The pair of "drinking wine" and "blood urine" was for example matched to the pair of "eating food" and "diarrhea", resulting in an expansion with the top terms of `food`, `pork, peanut, milk, health`.

The evaluation of retrieval results show that both expansion runs submitted by us do not improve the retrieval performance compared to the run solely using our custom reranking scheme. This might be due to a to high number of ill-fitting expansions for the *CauseNet* run and too long queries formulated by the ChatGPT expansion.

The scoring algorithm can be considered a basic measure. It works similarly to the query analysis method. Since the search corpus is much more complicated and varied than the task queries, we struggled to find many matches in the found documents. This means our proposal leans more towards a high-precision, low-recall solution. Still, the scoring lead to an improvement regarding the retrieval performance in relation to the other baseline run submitted.

References

[1] A. Bondarenko, M. Fröbe, J. Kiesel, F. Schlatt, V. Barriere, B. Ravenet, L. Hemamou, S. Luck, J. Reimer, B. Stein, M. Potthast, M. Hagen, Overview of Touché 2023: Argument and Causal Retrieval, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction. 14th International Conference of the CLEF Association (CLEF 2023), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2023, p. to appear.

[2] M. Fröbe, M. Wiegmann, N. Kolyada, B. Grahm, T. Elstner, F. Loebe, M. Hagen, B. Stein, M. Potthast, Continuous Integration for Reproducible Shared Tasks with TIRA.io, in: J. Kamps, L. Goeuriot, F. Crestani, M. Maistro, H. Joho, B. Davis, C. Gurrin, U. Kruschwitz, A. Caputo (Eds.), Advances in Information Retrieval. 45th European Conference on IR Research (ECIR 2023), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2023, pp. 236–241. URL: https://doi.org/10.1007/978-3-031-28241-6_20. doi:10.1007/978-3-031-28241-6_20.

[3] S. Heindorf, Y. Scholten, H. Wachsmuth, A. N. Ngomo, M. Potthast, CauseNet: Towards a Causality Graph Extracted from the Web, in: M. d'Aquin, S. Dietze, C. Hauff, E. Curry, P. Cudré-Mauroux (Eds.), CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020, ACM, 2020, pp. 3023–3030. URL: https://doi.org/10.1145/3340531.3412763. doi:10.1145/3340531.3412763.

[4] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, C. D. Manning, Stanza: A Python natural language processing toolkit for many human languages, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2020.

[5] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al., Improving language understanding by generative pre-training (2018).

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention Is All You Need, 2017. URL: http://arxiv.org/abs/1706.03762.

[7] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, CoRR abs/1810.04805 (2018). URL: http://arxiv.org/abs/1810.04805. arXiv:1810.04805.

[8] J. Bevendorff, B. Stein, M. Hagen, M. Potthast, Elastic ChatNoir: Search Engine for the

ClueWeb and the Common Crawl, in: L. Azzopardi, A. Hanbury, G. Pasi, B. Piwowarski (Eds.), Advances in Information Retrieval. 40th European Conference on IR Research (ECIR 2018), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2018.