Aspect-Based Academic Search Using Domain-Specific KB

Prajna Upadhyay[1(✉)], Srikanta Bedathur[1], Tanmoy Chakraborty[2], and Maya Ramanath[1]

[1] IIT Delhi, Hauz Khas, New Delhi 110016, India
{prajna.upadhyay,srikanta,ramanath}@cse.iitd.ac.in
[2] IIIT-Delhi, Okhla Industrial Estate, Phase III, New Delhi 110020, India
tanmoy@iiitd.ac.in

Abstract. Academic search engines allow scientists to explore related work relevant to a given query. Often, the user is also aware of the *aspect* to retrieve a relevant document. In such cases, existing search engines can be used by expanding the query with terms describing that aspect. However, this approach does not guarantee good results since plain keyword matches do not always imply relevance. To address this issue, we define and solve a novel academic search task, called *aspect-based retrieval*, which allows the user to specify the aspect along with the query to retrieve a ranked list of relevant documents. The primary idea is to estimate a language model for the aspect as well as the query using a domain-specific knowledge base and use a mixture of the two to determine the relevance of the article. Our evaluation of the results over the Open Research Corpus dataset shows that our method outperforms keyword-based expansion of query with aspect with and without relevance feedback.

Keywords: Academic retrieval · Aspect · Technical knowledge base

1 Introduction

Academic search engines such as Google Scholar, PubMed, and Semantic Scholar play a central role in the lives of researchers dealing with the ever growing flood of related work. To further improve the academic search experience, there have been proposals to either re-rank the results using user's interests [12] and the set of papers assessed relevant [2], or to recommend new articles based on a query article [4].

In this paper, we define and solve a novel academic search task, called *aspect-based retrieval*, which is targeted towards enabling the academic search user to specify the *aspect* along with the query to retrieve a ranked list of scientific articles that are (i) relevant to the query, and (ii) the relevance relation [3] between the query and retrieved documents is semantically close to the specified aspect. We illustrate this expected behavior with a concrete example: consider the query autoencoder and an aspect of interest, say, *application*, then we aim to rank high the articles which are related to the concept of autoencoder *and* are about the *applications* of autoencoders. If there were two papers, titled (a) "Complex-valued Autoencoders" and (b) "Exploring autoencoders for unsupervised feature selection", our system should rank the paper (b) higher than paper (a) since it specifically deals with applications of the specified query rather than its variants.

J. M. Jose et al. (Eds.): ECIR 2020, LNCS 12036, pp. 418–424, 2020.
https://doi.org/10.1007/978-3-030-45442-5_52

Note that both papers are indeed relevant to the central concept being queried. Aspect-based retrieval of scientific documents is not straightforward at all since the relation semantics do not manifest as simple keyword matches. Simply expanding the query with terms that define the aspect fails to retrieve relevant articles. In the above example, the paper (b) does not contain the application in the title as well as the abstract. On the other hand, a document, titled "Evaluating the Performance of Dynamic Database Applications" is not related to the query RDBMS along the *application* aspect although its title contains the term applications. The way in which the specified semantic relationship manifests between the query and a document is highly dependent on the domain we operate in. For example, given the query autoencoder, for a document to be related along *application* aspect, the presence of terms like feature or selection with terms similar to application would be highly suggestive because feature_selection is known to be an application of autoencoder. But this can not be easily determined by just analysing the documents without prior knowledge of the domain.

We address the challenge of aspect-oriented retrieval for scientific documents by minimizing the risk of returning a document whose language model diverges from the model estimated for domain-specific query and aspect specified by the user. The query and aspect models are derived using domain-specific knowledge bases (KBs), which is challenging in itself due to the inherent sparsity of relations in these KBs. By using a domain-specific KB of computer science, like TeKnowbase [14], we show how to overcome the sparsity issue in KB via inference using meta-paths derived from the KB. Our results over the Open Research Corpus [1] dataset containing more than 39 million published research papers show that our proposed approach outperforms variants of query likelihood language models with/without relevance feedback.

2 Aspect Based Retrieval Model

2.1 System Overview

Our system takes a query and an aspect as input, and returns a ranked list of relevant documents. Users express their information need as strings of words called **queries**. In our case, a query is a technical entity. An **aspect** is the relevance relation specified by the user between the query and the relevant document. Given a query, the documents are ranked using retrieval models [9]. A **retrieval model** transforms the document space into an intermediate representation and returns a ranked list of documents according to some scoring function.

2.2 Retrieval Model and Estimation

Language modelling techniques [10] model the relevance of a document as the probability of generating the query from the document. Expanding the query with aspect terms and doing relevance feedback [7] will only retrieve documents containing those terms. Given a query q and an aspect a, a relevant document d consists of terms determined by both q and a.

Aspect Dependent Prior Probability. $P(w|a)$ is the prior, which is the probability of a term w appearing in d given an aspect a, independent of the query.

Query and Aspect Dependent Probability. The probability of a term appearing in d given a query q and aspect a is denoted by $P(w|q, a)$.

Mixture of the Two Probability Distributions. The relevance of d will be determined by a mixture model of both the probability distributions. Equation (1) describes our probability distribution. It is denoted as MM.

$$MM(w) = \lambda P(w|a) + (1 - \lambda)P(w|q, a) \tag{1}$$

Scoring of Documents. The language model M_d of a candidate document is expressed by Eq. (2). Dirichlet smoothing is used for M_d.

$$M_d(w) = \frac{tf(w, d) + \mu P(w|C)}{length(d) + \mu} \tag{2}$$

where $tf(w, d)$ is the frequency of w in d, and $P(w|C)$ is the probability of w appearing in the entire collection. The risk associated with using MM to approximate M_d is expressed by KL-divergence between MM and M_d and the documents are returned in an order of increasing KL-divergence.

$$KL(MM||M_d) = \sum_w P(w|MM) \log \frac{P(w|MM)}{P(w|M_d)} \tag{3}$$

Estimation of Query-Independent Component. $P(w|a)$ is estimated using a narrow set of documents from our dataset acquired as follows. We chose 10 queries and retrieved the top 10 documents for them using the standard query likelihood model. Additionally, we fired queries of the form "query+aspect" to retrieve the top-10 documents using the same model. We recruited evaluators to annotate about 1500 documents with the aspect labels (described in details in Sect. 3.2). In order to increase the size of our document set, we used heuristics to collect more documents given an aspect. We formulated a query containing only the aspect as a keyword and retrieved documents for it using the standard likelihood model. But, we retained only those documents which contained the name of the aspect in the title on the intuition that such documents are highly likely (though not guaranteed) to be about those aspects. Having a set of ground-truth documents D, $P(w|a)$ is estimated according to Eq. (4).

$$P(w|a) = \frac{1}{|D|} \sum_{d \in D} \frac{tf(w, d)}{\sum_{w' \in d} tf(w', d)} \tag{4}$$

Estimation of Query-Dependent Component. We used relationships in TeKnowbase (TKB) [14] to represent aspects. TKB consists of entities such as hidden_markov_model or speech_recognition and other domain-specific relationships like application, implementation or algorithm. The triple ⟨`speech_recognition, application, hidden_markov_model`⟩ conveys information that speech_recognition is an application of hidden_markov_model. The entities connected via application relation in TKB have a higher probability of appearing in documents addressing *application* aspect. However, TKB is sparse. To automatically infer the entities participating in a particular relationship type, we used meta-paths. A meta-path is a sequence of edges with labels connecting two nodes which have been used previously for KB completion tasks [6], link prediction [8] as well as to find similarity between two nodes [11,13]. Figure 1 shows how meta-paths can be used to infer relationships between entities.

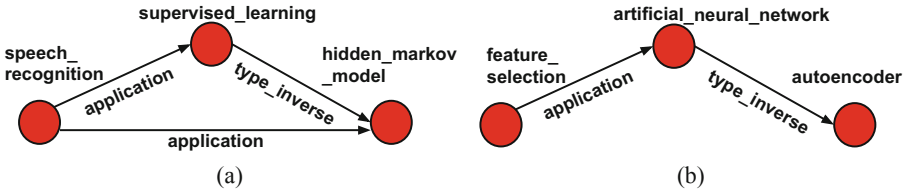

Fig. 1. Examples of meta-paths for application relation. (a) The meta-path ⟨application, type_inverse⟩ exists between speech_recognition and hidden_markov_model. (b) feature_selection and autoencoder are not related by application relation but still it can be inferred because of the existence of the same meta-path ⟨application, type_inverse⟩ between them.

(1) Direct inference using meta-paths. To automatically determine entities that participate in a given relationship type with entity e_i, we used the path-constrained random walk algorithm (PRA) proposed in [6]. Given a set E of entities in TKB, a source node e_i and a meta-path P, a path-constrained random walk defines a probability distribution $h_{e_i,P}(e_j)$ to all entities in E which is the probability of reaching e_j from e_i by doing a random walk along P. The key idea is to acquire the set of meta-paths representing the given relationship type and use PRA for inferencing. To do so, we retrieved the set of all meta-paths (MP) connecting the given relationship type in TKB and scored them according to their frequency. Given a set of meta-paths $MP = P_1, P_2, ..., P_n$, the score for each node reachable from source e_i is given by: $score_{e_i}(e_j) = \alpha_1 h_{e_i,P_1}(e_j) + \alpha_2 h_{e_i,P_2}(e_j) + ... + \alpha_n h_{e_i,P_n}(e_j)$, where $\alpha_l (l = 1 \ldots n)$ is the frequency of meta-path P_l. $score_{e_i}(e_j)$ is converted to a probability distribution using *softmax* and denoted by Eq. (5).

(2) Indirect inference using meta-paths. PRA assigns zero probability to nodes that are not reachable via any meta-paths in MP. To address this issue, we used MetaPath2Vec [5]. It takes a meta-path as input and constructs embeddings of entities such that entities that are likely to be connected via the meta-path (and not necessarily having a meta-path between them) are assigned vector representations closer to each other. We used the top-k meta-paths in MP as input to metapath2vec and obtained vector representations V_e for entity e. We used the softmax function to convert cosine similarities between entities into a probability distribution as described in Eq. (6).

$$DI_{e_i}(e_j) = \frac{e^{score_{e_i}(e_j)}}{\sum_{e_k \in |E|} e^{score_{e_i}(e_k)}} \tag{5}$$

$$h'_{e_i}(e_j) = \frac{e^{\mathrm{sim}(V(e_i),V(e_j))}}{\sum_{e_k \in |E|} e^{\mathrm{sim}(V(e_i),V(e_k))}}. \tag{6}$$

The probability distribution for inferencing is a mixture of $DI_{e_i}(e_j)$ and $h'_{e_i}(e_j)$ using β as given in Eq. (7). Since the documents are represented as bag of words, we defined the distribution over terms instead of entities using Eq. (8). $terms(e)$ is the set of words present in the entity e, and e_q is the entity that q represents. The final probability is a mixture given by Eq. (1).

$$P_a(e_i|e_j) = \beta * DI_{e_i}(e_j) + (1 - \beta) * h'_{e_i}(e_j) \tag{7}$$

$$P(w|q, a) = \sum_e P_a(e|e_q), \text{s.t. } w \in terms(e) \tag{8}$$

3 Experiments

3.1 Setup

Dataset. We used the Open Research Corpus dataset and indexed it using Galago. The baseline models (described below) are already implemented in Galago.

Aspects. We experimented with 3 different aspects – *application*, *algorithm* and *implementation*. We set λ and β to 0.5 in Eqs. (1) and (7). We restricted ourselves to meta-paths of size at most 3. We set $k = 5$ for choosing the top-k meta-paths for generating embeddings using MetaPath2Vec (described in Sect. 2.2).

Benchmarks. Benchmark queries were taken from a set of 100 queries released by [15] out of which 43 existed as whole entities in TKB, shown in Fig. 2.

artificial intelligence, augmented reality, autoencoder, big data, category theory, clojure, cnn, computer vision, cryptography, data mining, data science, deep learning, differential evolution, dirichlet process, duality, genetic algorithm, graph drawing, graph theory, hashing, information geometry, information retrieval, information theory, knowledge graph, machine learning, memory hierarchy, mobile payment, natural language, neural network, ontology, personality trait, prolog, question answering, recommender system, reinforcement learning, sap, semantic web, sentiment analysis, smart thermostat, social media, speech recognition, supervised learning, variable neighborhood search, word embedding

Fig. 2. Benchmark queries

Baselines. We explicitly added the keyword representing the aspect to the query and used standard retrieval models with/without relevance feedback techniques as baselines described below:

(1) Query likelihood model with query only (QL+query). Query likelihood [10] estimates a language model for each document in the collection and ranks them by the likelihood of seeing the query terms as a random sample given that document model.
(2) Query likelihood model with query + aspect name (QL + query + aspect). We used the same model as above but added the terms application, algorithm or implementation to the query based on the aspect and retrieved the results.
(3) Query expansion with pseudo relevance feedback on QL + query + aspect (QL + query + aspect + QE). We chose top-100 terms to expand the query for the query used in the previous baseline using relevance feedback model [7]. Top-1000 documents were used as feedback documents. The weight of the original query was set as 0.75.
(4) Mixture Model (MM). This is our retrieval model described in Sect. 2.

3.2 Evaluation Scheme and Metrics

Evaluation Scheme. In the absence of an extensive ground-truth dataset, we conducted a crowd-sourced user-evaluation exercise (involving Computer Science students and researchers, not related to the project) to measure the performance of our model. We formulated domain-specific questions, and depending on the answers marked by the evaluators, the documents were assigned a particular score for a query and aspect pair.

Evaluation Metrics. Each query and abstract pair was graded by at least 2 evaluators. We converted the response from each of them into a graded relevance scale and averaged the relevance values marked by them for each query-abstract pair. We used **Discounted Cumulative Gain (DCG)** and **Precision** to evaluate top-5 documents.

3.3 Results and Discussions

Table 1 shows the results for *algorithm*, *application* and *implementation* aspects. We observe that our model outperforms the rest of the baselines in terms of precision@5 and DCG for all of the 3 aspects. **QL + query + aspect + QE** comes second after our retrieval model. By modelling the aspect and query dependant probability explicitly, we were able to address the problems of simple keyword-based match for aspects described in Sects. 1 and 2. For example, the top-2 papers retrieved for genetic_algorithm for application aspect by our model were *Genetic Ant Algorithm for Continuous Function Optimization and Its MATLAB Implementation* and *Solve Zero-One Knapsack Problem by Greedy Genetic Algorithm*. The top-2 papers retrieved by **QL + query + aspect + QE** for *application* aspect do not describe any application of genetic_algorithm but contained a few terms like "a wide application prospect" in the abstract due to which it was retrieved in the top positions. Adding relevance feedback terms also did not work because the list of pseudo relevant documents did not contain relevant documents in the first place due to plain keyword-based retrieval. Both the papers retrieved by our method address application aspect for genetic_algorithm even if "application" is not mentioned in the title.

Table 1. Results for algorithm, application and implementation aspect.

Approach	Algorithm			Application			Implementation		
	DCG@5	P@5	P@1	DCG@5	P@5	P@1	DCG@5	P@5	P@1
MM	**6.27**	**0.70**	**0.75**	**2.64**	**0.45**	**0.47**	**2.33**	**0.44**	0.40
QL+query	2.69	0.3	0.33	1.42	0.25	0.22	1.05	0.16	0.23
QL+query+aspect	5.03	0.56	0.59	2.38	0.41	0.35	1.92	0.30	*0.43*
QL+query+aspect+QE	*5.12*	*0.58*	*0.61*	*2.5*	*0.43*	*0.41*	*2.29*	*0.37*	**0.49**

4 Conclusion

In this paper, we built an aspect-based retrieval model for scientific literature using TeKnowbase. Given a query and an aspect, this model returns a ranked list of documents that address that aspect for the query. We tested our model for 43 queries and 3 aspects with satisfactory results. We could beat the results obtained by adding aspect name explicitly to the query and doing pseudo-relevance feedback on those documents.

Acknowledgements. This work was partially supported by IIT Delhi-IBM Research AI Horizons Network collaborative grant; Ramanujan Fellowship, DST (ECR/2017/00l691) and the Infosys Centre for AI, IIIT-Delhi, India. T. Chakraborty would like to thank the support of Google India Faculty Award.

References

1. Ammar, W., et al.: Construction of the literature graph in semantic scholar. In: NAACL (2018)
2. Raamkumar, A.S, Foo, S., Pang, N.: Can I have more ofthese please?: assisting researchers in finding similar research papers from a seed basket of papers. The Electronic Library (2018)
3. Bean, C., Green, R.: Relevance Relationships. Information Science and Knowledge Management (2001)
4. Chakraborty, T., Krishna, A., Singh, M., Ganguly, N., Goyal, P., Mukherjee, A.: FeRoSA: a faceted recommendation system for scientific articles. In: PAKDD (2016)
5. Dong, Y., Chawla, N.V., Swami, A.: Metapath2Vec: scalable representation learning for heterogeneous networks. In: KDD (2017)
6. Lao, N., Mitchell, T., Cohen, W.W.: Random walk inference and learning in a large scale knowledge base. In: EMNLP (2011)
7. Lavrenko, V., Croft, W.B.: Relevance based language models. In: SIGIR (2001)
8. Ley, M.: DBLP - some lessons learned. In: PVLDB (2009)
9. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)
10. Ponte, J.M., Croft, W.B.: A language modeling approach to information retrieval. In: SIGIR (1998)
11. Shi, C., Kong, X., Yu, P.S., Xie, S., Wu, B.: Relevance search in heterogeneous networks. In: EDBT (2012)
12. Sugiyama, K., Kan, M.-Y.: A comprehensive evaluation of scholarly paper recommendation using potential citation papers. Int. J. Digit. Libr. **16**(2), 91–109 (2014). https://doi.org/10.1007/s00799-014-0122-2
13. Sun, Y., Barber, R., Gupta, M., Aggarwal, C.C., Han, J.: Co-author relationship prediction in heterogeneous bibliographic networks. In: ASONAM (2011)
14. Upadhyay, P., Bindal, A., Kumar, M., Ramanath, M.: Construction and applications of teknowbase: a knowledge base of computer science concepts. In: Companion Proceedings of the The Web Conference (2018)
15. Xiong, C., Power, R., Callan, J.: Explicit semantic ranking for academic search via knowledge graph embedding. In: WWW (2017)