Construction and Applications of TeKnowbase

Prajna Upadhyay
prajna.upadhyay@cse.iitd.ac.in
Indian Institute of Technology, Delhi
New Delhi-110016, India

Maya Ramanath
ramanath@cse.iitd.ac.in
Indian Institute of Technology, Delhi
New Delhi-110016, India

1 INTRODUCTION

With advances in information extraction research, and the availability of large amounts of structured and unstructured (textual) data, *automatic* construction of knowledge-bases are not only possible, but also desirable because of the coverage they can offer. There are already many such general-purpose knowledge-bases such as Yago [8] and DBPedia [5]. However, due to the lack of such knowledge bases for domains such as Computer Science, it becomes critical to revisit the automatic construction processes that take advantage of domain-specific resources. A knowledge base in the domain of Computer Science opens up possibilities for designing new academic applications as well as improving the ones that already exist, by complementing them with knowledge, which is not readily available in textual sources. For example, such knowledge graphs can be used to improve the academic search experience in scholarly data retrieval as well as imparting Computer Science education.

To address this gap, we constructed TeKnowbase[1] [9, 11], which stores triples from the domain of Computer Science in the RDF format. For e.g. the fact that "Activity recognition is an application of hidden markov models" can be stored as a triple ⟨`activity_recognition, application, hidden_markov_model`⟩. We then used TeKnowbase to design two novel applications to assist Computer Science enthusiasts in scholarly data retrieval. The construction of TeKnowbase and these applications are described in the following sections.

1.1 Construction of TeKnowbase

The construction of TeKnowbase is described in details in [9]. We first extracted a list of entities from Wikipedia as well as domain-specfic websites like Webopedia and Techtarget, and then used heuristics, Open IE [1] and inferencing techniques [7] to extract relationships between them. We used humans to evaluate a subset of the extracted triples. We also showed that using features from TeKnowbase improved the classification of Stack Overflow posts, ranking of research papers and retrieval of pre-requisites [2, 6].

1.2 TeKnowbase for aspect-based retrieval of research papers

We used TeKnowbase to assist a user in *aspect-based retrieval* of research papers, which is a novel application [10] that takes a query and an aspect as input and returns a ranked list of documents, relevant to both the query and the aspect. An academic search user, who has some idea about the domain, often also has an *aspect* of interest in mind along with the query she is interested in. For example, a scholar might be interested in the *applications* of genetic_algorithm. In such a scenario, the query is genetic_algorithm and the aspect is *application*. Such a system should rank a paper titled "Multi-Objective Genetic Algorithm for Robust Clustering with Unknown Number of Clusters" higher than "Hierarchical Distributed Genetic Algorithms" because the former describes an application of genetic_algorithm. Retrieving papers both relevant to the query and the aspect is challenging because plain keyword matches do not always imply relevance, and the relevance is highly dependant on the domain knowledge. Aspect-based retrieval uses language models to represent the query and documents estimated using TeKnowbase and ranks them according to risk minimization framework. Our evaluation over the Open Research Corpus[2] consisting of more than 29 million abstracts shows that our model outperforms variants of query likelihood model such as pseudo-relevance feedback as well as state-of-the art diversification and neural models.

1.3 TeKnowbase for faceted retrieval of pre-requisites

A student who has just joined the PhD programme will face difficulty understanding certain concepts, for which she does not have the required *pre-requisite* knowledge [2], [6]. She can search for the query on the web but there is no guarantee that the retrieved documents will contain its pre-requisites and even if they do, she may need to further refer to the pre-requisite's pre-requisite. This leads to a chain of searches and is time-consuming for her. It would be helpful to have a retrieval system that automatically generates pre-requisites for the concept of interest. Not only that, organizing the pre-requisites into multiple facets [3, 4] will further help with an overall understanding of the concept. For example, to understand conditional_random_field, along with the knowledge of conditional_probability, a knowledge of named_entity_recognition will help her understand *applications* of the queried concept, because named_entity_recognition is an application of conditional_random_field. To address these issues, we developed *PreFace*, which is a retrieval model to extract faceted pre-requisites for queries. PreFace solves the problem of facet extraction and pre-requisite determination together, because solving them separately does not return good results. This is because i) existing facet extraction techniques [3, 4] use open domain knowledge bases, which are not suitable to extract facets for a query in the domain of Computer Science, so domain-specific KBs like TeKnowbase have to be used ii) domain-specific KBs like TeKnowbase are sparse, so existing techniques fail to retrieve facets from them, which assume that the knowledge base generates a large number of facets and focus on their efficient ranking. To solve this problem, PreFace uses key-phrases extracted from relevant research papers for the query as candidate facets and uses a language model representation for them. This language model is estimated using TeKnowbase and the facets are ranked according to a probabilistic framework, balancing the relevance and the diversity of the retrieved facets. Our evaluation of results over a standard benchmark set of queries shows that Preface retrieves better facets and pre-requisites than state-of-the art techniques.

[1] Dataset freely available from https://github.com/prajnaupadhyay/TeKnowbase

[2] https://allenai.org/data/s2orc

REFERENCES

[1] Michele Banko et al. 2007. Open Information Extraction from the Web. *IJCAI* (2007), 2670–2676.

[2] Chen Liang and others. 2015. Measuring prerequisite relations among concepts. *EMNLP* (2015).

[3] Leila Feddoul et al. 2019. Automatic Facet Generation and Selection over Knowledge Graphs. *SEMANTICS* (2019).

[4] Z. Jiang, Z. Dou, and J. Wen. 2017. Generating Query Facets Using Knowledge Bases. *IEEE Transactions on Knowledge and Data Engineering* 29, 2 (2017), 315–329.

[5] Jens Lehmann et al. 2015. DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web* 6, 2 (2015), 167–195.

[6] Irene Li et al. 2019. What Should I Learn First: Introducing LectureBank for NLP Education and Prerequisite Chain Learning. *AAAI* (2019).

[7] Richard Socher et al. 2013. Reasoning with Neural Tensor Networks for Knowledge Base Completion. *NIPS* (2013).

[8] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago - a core of semantic knowledge. *WWW* (2007).

[9] Prajna Upadhyay et al. 2018. Construction and Applications of TeKnowbase: A Knowledge Base of Computer Science concepts. *WWW Companion* (2018).

[10] Prajna Upadhyay et al. 2020. Aspect-Based Academic Search Using Domain-Specific KB. *ECIR* (2020).

[11] Prajna Upadhyay, Tanuma Patra, Ashwini Purkar, and Maya Ramanath. 2016. TeKnowbase: Towards Construction of a Knowledge-base of Technical Concepts. *Technical report available from http://arxiv.org/abs/1612.04988* (2016).