Data & Knowledge Engineering 107 (2017) 24–50

Contents lists available at ScienceDirect

Data & Knowledge Engineering

journal homepage: www.elsevier.com/locate/datak

Matching parse thickets for open domain question answering

Boris Galitsky

Knowledge-Trail Inc., 371 Cureton Pl, San Jose, CA 95127, USA

ARTICLE INFO

Keywords:
Question-answering systems
Discourse analysis
Rhetoric structure
Parse tree
Parse thicket

ABSTRACT

Traditional parse trees are combined together and enriched with anaphora and rhetoric information to form a unified representation for a paragraph of text. We refer to these representations as *parse thickets*. They are introduced to support answering complex questions, which include multiple sentences, to tackle as many constraints expressed in this question as possible. The question answering system is designed so that an initial set of answers, which is obtained by a TF*IDF or other keyword search model, is re-ranked. Passage re-ranking is performed using matching of the parse thickets of answers with the parse thicket of the question. To do that, a graph representation and matching technique for parse structures for paragraphs of text have been developed. We define the operation of generalization of two parse thickets as a measure of the distance between paragraphs of text to be the maximal common sub-graph of these parse thickets. A partial case of parse thickets*, a rhetoric map of an answer*, allows leveraging discourse for relevance in a rule-based manner.

Passage re-ranking improvement via parse thickets is evaluated in a variety of search domains with long questions. Using parse thickets improves search accuracy compared with the bag-of words, the pairwise matching of parse trees for sentences, and the tree kernel approaches. As a baseline, we use a web search engine API, which provides much more accurate search results than the majority of search benchmarks, such as TREC. A comparative analysis of the impact of various sources of discourse information on the search accuracy is conducted. An open source plug-in for SOLR is developed so that the proposed technology can be easily integrated with industrial search engines.

1. Introduction

According to Noam Chomsky, “the fundamental aim in the linguistic analysis of a language is to separate the grammatical sequences which are the sentences of a language, from the ungrammatical sequences, which are not sentences of this language, and to study the structure of the grammatical sequences.” Parse trees have become a standard form for representing these grammatical sequences, to represent their syntactic structures [20,57]. Such representation is essential for structured comparisons of sentences; it also enriches the feature set of learning. However, there is no generally accepted structure at the level of a text paragraph that would play a similar role. Such a paragraph-level model needs to involve a set of parse trees for each sentence of the paragraph and the paragraph-level discourse information. HowWe refer to the sequence of parse trees plus a number of arcs for inter-sentence relations of the discourse type between the nodes for words as a *parse thicket*. It is a graph that includes parse trees for each sentence, as well as additional arcs for inter-sentence discourse relationships. In our earlier studies, development of the parse thickets representation was stimulated by the task of comparing two paragraphs of text in a way that is invariant to how the information is divided among sentences. In this study, we explore how parse thickets of questions and answers can be matched.

It is hard for web search engines to handle fairly long queries consisting of multiple sentences because it is unclear which keywords are more important and which are less important. In most cases, being fed with multi-sentence queries, search engines such as Google and Bing deliver either very similar, almost duplicate documents or search results very dissimilar to the query,

http://dx.doi.org/10.1016/j.datak.2016.11.002
Received 19 August 2013; Received in revised form 19 October 2015; Accepted 30 November 2016
Available online 09 December 2016

leaving almost all keywords unmatched. This happens because it is difficult to learn user clicks–based ranking in a higher-dimensional case for more than ten keywords (the number of longer queries is fairly high). Hence, modern search engines require a technique that orders potential answers based on minimization of their structural distance from the question. This can be performed by applying graph-based representations of both question and answer so that one can match not only parse trees with questions and answers but also their entire discourse.

The demand for access to different types of information has led to a renewed interest in answering questions posed in ordinary human language and seeking exact, specific and complete answers. After having made substantial achievements in fact-finding and list questions, the natural language processing (NLP) community turned their attention to more complex information needs that cannot be answered by simply extracting named entities (persons, organization, locations, dates, etc.) from single sentences in documents [11]. Unlike simple fact-finding queries, complex questions include multiple sentences; therefore, their keywords should not be matched all together to produce a meaningful answer: the query representation must take its discourse information into account.

Most web search engines first attempt to find the occurrence of query keywords in a single sentence, and when that is not possible or has a low TF*IDF score, a set of keywords may be accepted spreading through more than one sentence. The indices of these search engines have no means to keep information on whether found occurrences of the query keywords in multiple sentences are related to one another, to the same entity, and, being in different sentences, are all related to the query term. Once a linguistic representation goes beyond a bag-of-words, the necessity arises for a systematic way to compare such representations, which go beyond sentence boundaries for questions and answers. However, there is a lack of formal models for comparing linguistic structures. In this paper we propose a mechanism for building a required search index that contains discourse-level constraints to improve search relevance.

Answering complex questions with keywords distributed through distinct sentences of a candidate answer is a sophisticated problem requiring deep linguistic analysis. If the question keywords occur in different sentences of an answer in a linguistically connected manner, this answer is most likely relevant. This is usually true when all of these keywords occur in the same sentence; then, they should be connected syntactically. For the inter-sentence connections, these keywords need to be connected via anaphora, refer to the same entity or sub-entity, or be linked by rhetoric discourse.

If the question keywords occur in different sentences, there should be linguistic cues for some sort of connection between these occurrences. If there is no connection between the question keywords in an answer, then different constraints for an object expressed by a question may be applied to different objects in the answer text, and this answer is therefore irrelevant to this question.

The main classes of discourse connections between sentences are as follows:

- Anaphora. If two areas of keyword occurrences are connected with an anaphoric relation, the answer is most likely relevant.
- Communicative actions. If a text contains a dialogue, and some question keywords are in a request and others are in the reply to this request, then then these keywords are connected and the answer is relevant. To identify such situations, one needs to find a pair of communicative actions and to confirm that this pair is of *request-reply* type.
- Rhetoric relations. These indicate the coherence structure of a text (Mann et al. [49]). Rhetoric relations for text can be represented by a discourse tree (DT), which is a labeled tree in which the leaves of the tree correspond to contiguous units for clauses (elementary discourse units, EDUs). Adjacent EDUs, as well as higher-level (larger) discourse units, are organized in a hierarchy by rhetoric relation (e.g., *background, attribution*). An anti-symmetric relation involves a pair of EDUs: nuclei, which are core parts of the relation, and satellites, which are the supportive parts of the rhetoric relation.

The most important class of discourse connection between sentences that we focus on in this study is *rhetoric*. Once an answer text is split into EDUs and rhetoric relations are established between them, it is possible to establish rules for whether query keywords occurring in the text are connected by rhetoric relations (and therefore this answer is likely relevant) or not connected (and this answer is most likely irrelevant). Hence, we use the DT so that certain sets of nodes in the DT correspond to questions where this text is a valid answer and certain sets of nodes correspond to an invalid answer.

In our earlier studies [24,21] we applied graph learning to parse trees at the levels of both sentences and paragraphs; here we proceed to the structured graph-based match of parse thickets. Whereas for text classification problems, learning is natural, it is not obvious how one can learn by answering a given question, given a training set of valid and invalid question-answer pairs. It would only be possible either in a very narrow domain or by enforcing a particular commonality in the question-answer structure, which has a very limited value. The aim of this study is to support the domain-independent answering of complex questions where keyword statistics possibilities are very restricted.

We have defined the least general generalization of parse trees (we call it syntactic generalization), and in this study, we extend it to parse thickets. We have applied generalizations of parse trees in search scenarios where a query is based on a single sentence and candidate answers come from single sentences [21] and multiple sentences [22]. In these cases, to re-rank answers, we needed pair-wise sentence-sentence generalization and sentence-paragraph generalizations. In this study we rely on parse thickets to perform a paragraph-level generalization, where both questions and answers are paragraphs of text. Whereas a number of studies applied machine learning techniques, such as convolution kernels ([34] and [52]) and syntactic parse trees [14], learning paragraph-level representation is an area to be explored.

The paper is organized as follows. We demonstrate the necessity for using discourse-level analysis to answer complex questions. We then introduce a generalization of parse thickets for question and answer as an operation that performs a relevance assessment. Generalization operation occurs at the level of words, phrases, rhetoric relations, communicative actions, sentences, and paragraphs. Tree kernels for parse thickets are defined so that we can compare them to direct graph matching. We then proceed to a simpler case