Technical Perspective: Constant-Delay Enumeration for Nondeterministic Document Spanners

Benny Kimelfeld
Technion, Israel
bennyk@cs.technion.ac.il

The challenge of extracting structured information from text, or sequential data in general, is prevalent across a multitude of data-science domains. This challenge, known as Information Extraction (IE), instantiates to core components in text analytics, and a plethora of IE paradigms have been developed over the past decades. Rules and rule systems have consistently been key components in such paradigms, yet their roles have varied and evolved over time. Analytics engines such as IBM's SystemT use IE rules for materializing relations inside *relational query languages*. Machine-learning classifiers and probabilistic graphical models (e.g., Conditional Random Fields) use rules for *feature generation*. They also serve as *weak constraints* in Markov Logic Networks (and extensions such as DeepDive), and generators of noisy *training data* in the state-of-the-art Snorkel system.

Originally introduced as the theoretical basis underlying SystemT [5], the framework of *document spanners* provides an abstraction for IE rules [2]. A document spanner states how a document is translated into a relation over its spans. More formally, a *document* is a string **d** over a finite alphabet, a *span* of **d** represents a substring of **d** by its start and end positions, and a *document spanner* is a function that maps every document **d** into a relation over the spans of **d**. The most studied class of document spanners is that of the *regular spanners*—the closure of regular expressions with capture variables under the operators of the Relational Algebra (RA): projection, natural join, union, and difference. Equivalently, the regular spanners are the ones expressible as *Variable-set Automata* (VAs)—nondeterministic finite-state automata that can open and close capture variables.

Past research on document spanners has focused on two main facets: *expressiveness*—which queries can be answered by combining basic text matchers with relational operators? and *complexity*—what is the computational gain of the holistic treatment of the combination, as opposed to the direct way of evaluating relational queries over materialized matchings? The paper "Constant-Delay Enumeration for Nondeterministic Document Spanners" by Amarilli, Bourhis, Mengel and Niewerth [1] makes a substantial leap in our understanding of the second facet.

Prior studies analyzed the complexity of regular spanners under two yardsticks of efficiency. Freydenberger, Kimelfeld, and Peterfreund [4] showed how to compile queries into VAs, and how to evaluate VAs with *polynomial delay in combined complexity*, where both the query and the document are considered input. Florenzano et al. [3] gave algorithms for evaluating VAs in *constant delay in data complexity*, following a linear-time preprocessing phase; this means that, fixing the VA, the evaluation time is (asymptotically) what it takes just to read the document and print the answers one by one. Interestingly, in the first case [4] the delay is inherently dependent on the document size, and in the second case [3] the delay is inherently exponential in the VA size.

While one could suggest that we need to choose between the two yardsticks of efficiency, Amarilli et al. [1] present an algorithm that, surprisingly, delivers both guarantees *at the same time*: in data complexity, their algorithm enumerates with a constant delay following linear-time preprocessing, and in addition, all time intervals are polynomial in the size of the VA. The algorithm is nontrivial, yet quite elegant. In constant-delay algorithms, the crux is typically in the data structure constructed in the preprocessing phase. Here, this data structure is the *mapping DAG* that provides a decision-diagram-like compact representation of the space of answers. In this work, however, a considerable part of the sophistication comes from way that this structure is used at the (constant-delay) enumeration phase. The general idea seems to be useful well beyond the scope of the paper. Importantly, the algorithm has also been implemented and released as open-source in Rust.[1]

1. REFERENCES

[1] A. Amarilli, P. Bourhis, S. Mengel, and M. Niewerth. Constant-delay enumeration for nondeterministic document spanners. In *ICDT*, pages 22:1–22:19, 2019.

[2] R. Fagin, B. Kimelfeld, F. Reiss, and S. Vansummeren. Document spanners: A formal approach to information extraction. *Journal of the ACM*, 62(2):12:1–12:51, 2015.

[3] F. Florenzano, C. Riveros, M. Ugarte, S. Vansummeren, and D. Vrgoc. Constant delay algorithms for regular document spanners. In *PODS*, pages 165–177. ACM, 2018.

[4] D. D. Freydenberger, B. Kimelfeld, and L. Peterfreund. Joining extractions of regular expressions. In *PODS*, pages 137–149. ACM, 2018.

[5] R. Krishnamurthy, Y. Li, S. Raghavan, F. Reiss, S. Vaithyanathan, and H. Zhu. SystemT: A system for declarative information extraction. *SIGMOD Record*, 37(4):7–13, 2008.

[1] `https://github.com/PoDMR/enum-spanner-rs`