

Information Extraction

Information extraction is the concept that involves structuring and combining data, whether it is explicitly stated or inferred, in texts. For a rule language, its value and tradeoffs can be analyzed by two concepts: expressivity and simplicity. Whether expressivity allows greater quality on the results and simplicity allows better human understandability, higher throughput and better performance to train models. Nevertheless, the lack of formalisms to abstract and define the expressivity, has made it difficult to analyze and compare the rule languages.

Document Spanners

In order to solve the lack of formalisms, a framework called document spanners has been defined by leveraging the principles of database management and setting up mathematical formulas to abstract a rule language.

A document spanner models a program for Information Extraction (IE) as a function that takes as input a text document (string over a finite alphabet) and produces a relation of spans (intervals in the document) over a predefined schema.

String: Jon Doe is an artist.

$\Upsilon_{st} = \Sigma^* \cdot x \{ \text{Jon} \vee \text{Doe} \} \cdot \Sigma^*$

Spans: $[1,3\rangle$ $[5,7\rangle$

State-of-the-Art	
Document spanners on SpLog	Decision problems performance
Freydenberger proposes Subword Property Logic (SpLog), a subset of EC^{REG} that has the same expressive power as any of the originally proposed core spanners and define methods to transform from core spanner logic to SpLog. .	Freydenberger and Holldack compared the performance of document spanners against patterns (word consisting of variables and terminals), word equations (equations of the form $x = y$, where x and y are patterns), and regular expressions (which allow repetition operators). They found out that string equality and join operators are really expensive and thus that some techniques will need to be developed to restrict their use. This will lead to the creation of subclasses of document spanners. .
Coupling ontologies with document spanner	Recursive programs for document spanners
Lembo and Scafoglieri introduce the concept of linking document spanners with existing ontologies following the Ontology-based Data Access (OBDA) framework. Such framework provides users with access to the information in their data sources through a three-level architecture, constituted by the ontology, the sources, and the mappings between the two. The Ontology-based Document Spanning (OBDS) system creates the mappings for the OBDA, associating the document spanner to the conjunctive query (CQ). Furthermore, they do not only introduce the concept, but they provided an algorithm that rewrites every CQ issued over a DL-Lite _R OBDS System into a spanner through means of the GLAV schema .	Peterfreund et al. go beyond regular spanners (spanners built on top of non-recursive Datalog and regular expressions) and develop the spanner language RGXlog which is made utilizing recursive Datalog over regular expressions. They found out four main things: 1) RGXlog is more expressible than the originally proposed core spanners (since it can tell whether or not two spans have the same length), 2) it is possible to write a program to determine whether or not two spans are the same string (avoiding the necessity to include the string-equality predicate), 3) that every RGXlog program can be evaluated in polynomial time (making RGXlog capable of expressing all polynomial time document spanners), and finally 4) that they only need to express their regex formulas utilizing two variables in order to make it polynomial time. .