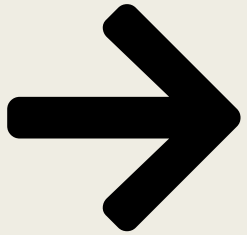
SPLIT-CORRECTNESS IN INFORMATION EXTRACTION

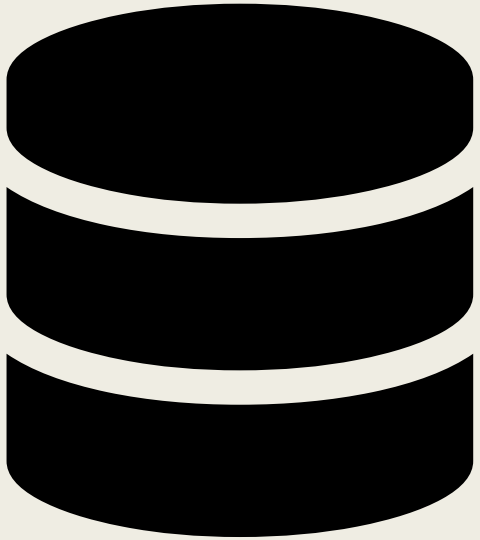
Johannes Doleschal - Benny Kimelfeld – Wim Martens
Yoav Nahshon - Frank Neven

Motivation

Information Extraction

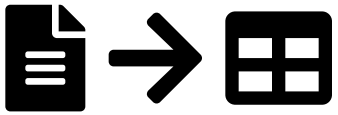
Input: Huge text documents, e.g. from server logs or financial reports
Task: Convert the data into a structured relational format





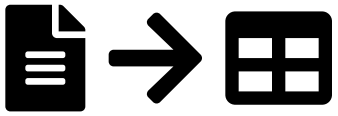
Information Extraction

Naive Extraction

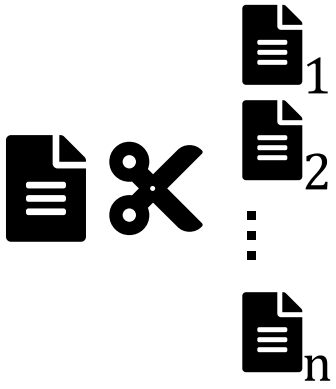


Information Extraction

Naive Extraction



Parallel Extraction

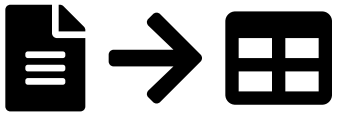


Possible splits ✂:

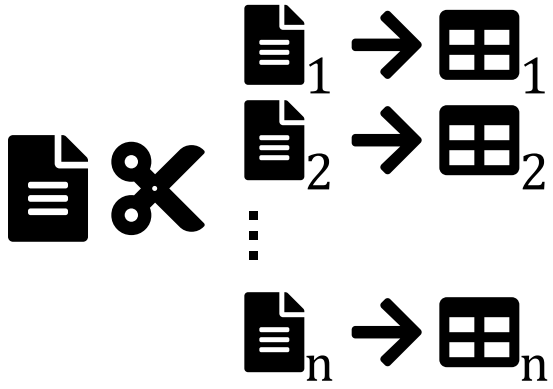
- Text documents into paragraphs or sentences,
- error logs into exceptions,
- server logs into HTTP messages, …

Information Extraction

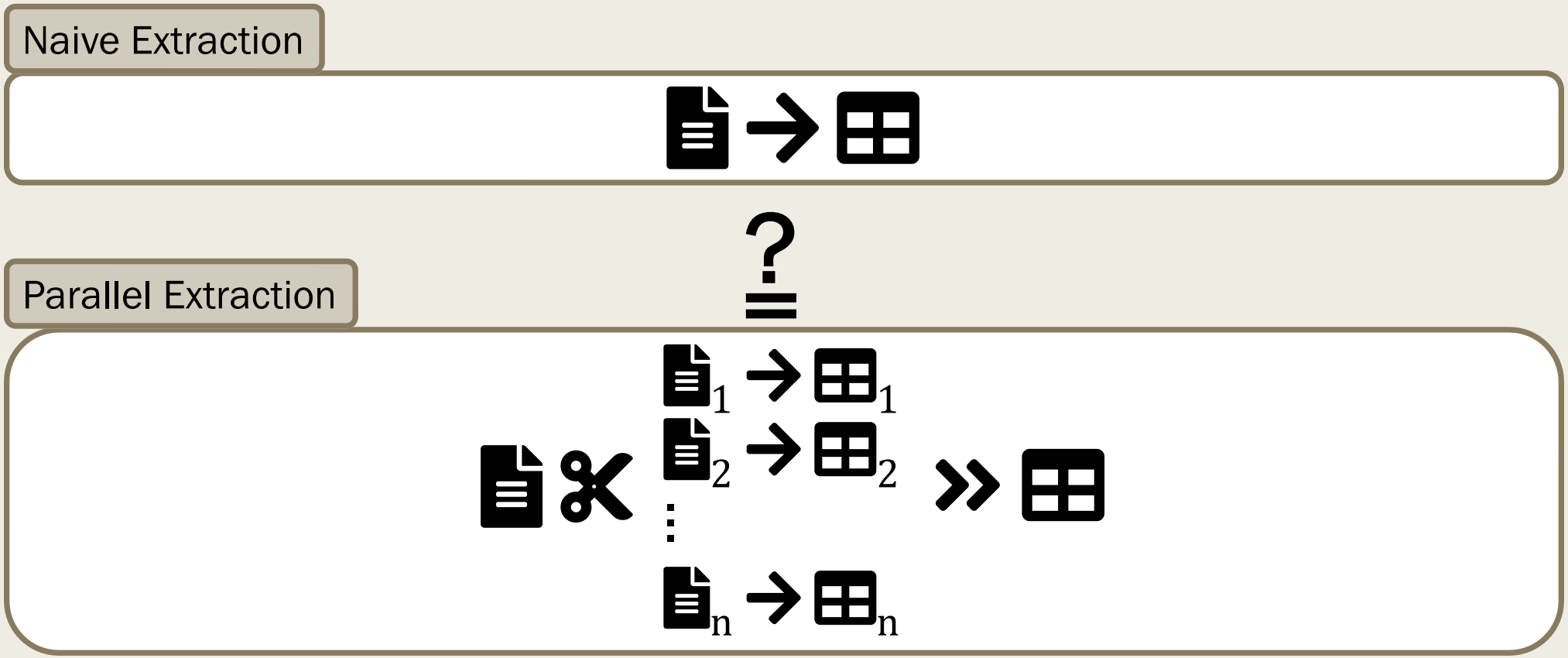
Naive Extraction



Parallel Extraction



Information Extraction



More Formally

Information Extractor →

$$\rightarrow(\blacksquare) = \boxplus$$

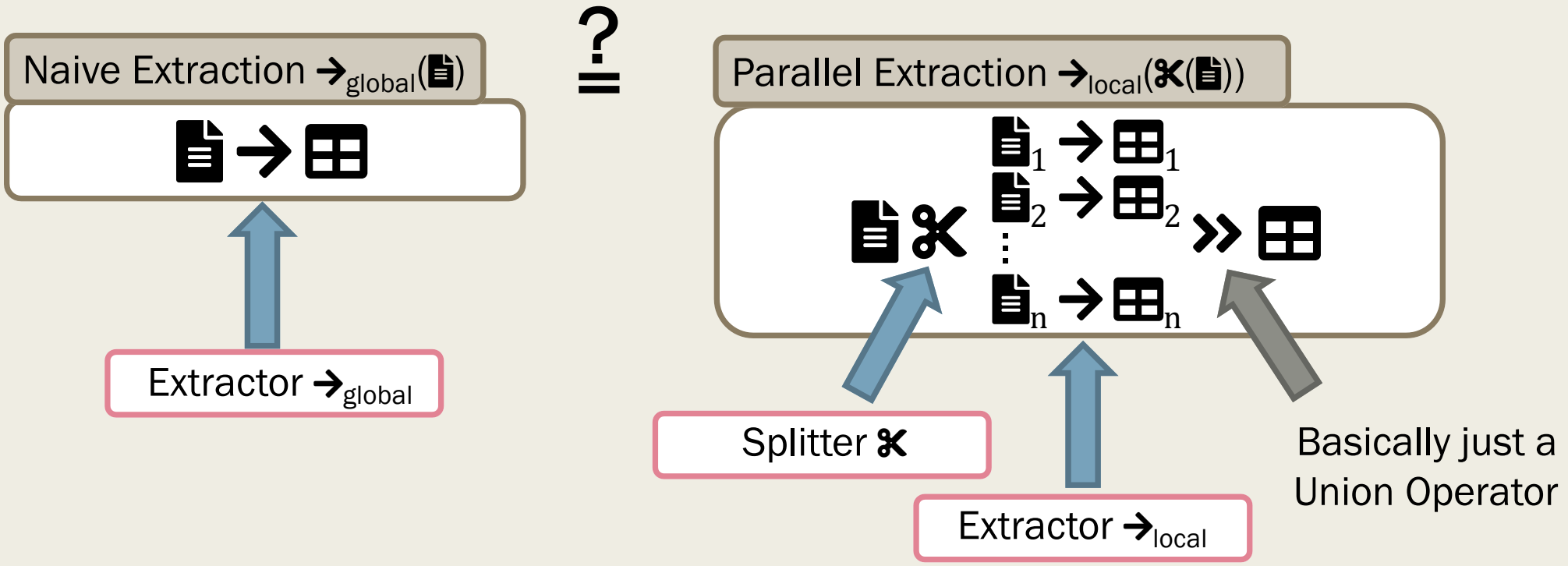
Splitter ✂

$$\text{✂}(\blacksquare) = \blacksquare_1, \ldots, \blacksquare_n$$

Evaluating Extractor → on Splitter ✂

$$\rightarrow(\text{✂}(\blacksquare)) = \bigcup_{1 \le i \le n} \rightarrow(\blacksquare_i) = \boxplus$$

Central Question

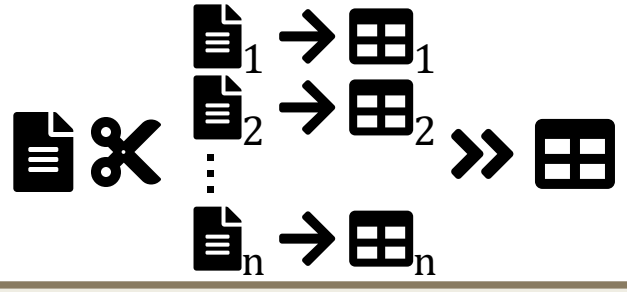


Main Problems

Naive Extraction $\rightarrow_{global}(d)$

$\stackrel{?}{=}$

Parallel Extraction $\rightarrow_{local}(\kappa(d))$



Split-Correctness

Input: Information Extractors $\rightarrow_{global}$, $\rightarrow_{local}$, Splitter κ

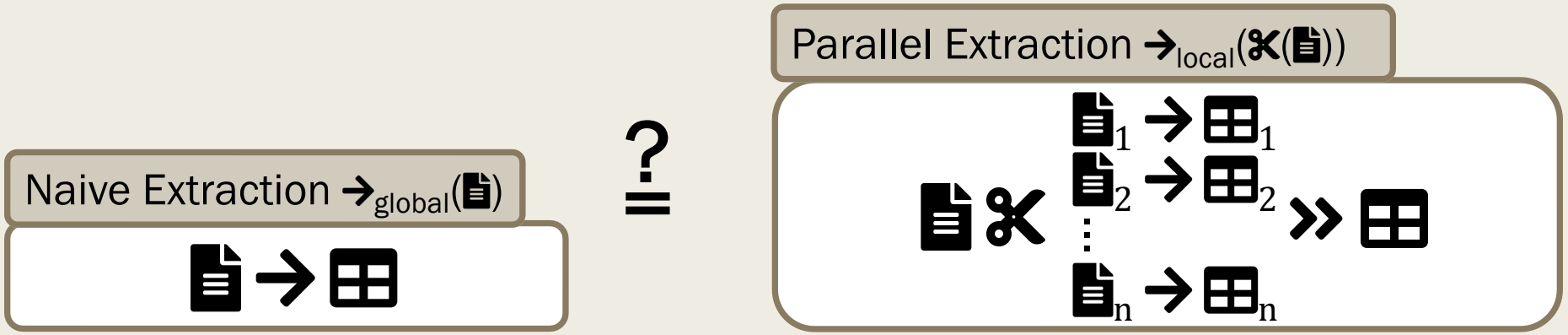
Question: Is $\rightarrow_{global}(d) = \rightarrow_{local}(\kappa(d))$, for all documents d?

Self-Splittability

Input: Information Extractor $\rightarrow$, Splitter κ

Question: Is $\rightarrow(d) = \rightarrow(\kappa(d))$, for all documents d?

Main Problems



Splittability

Input: Information Extractor →$_{global}$, Splitter ✂

Question: Is there an Information Extractor →$_{local}$, such that →$_{global}$(📄) = →$_{local}$(✂(📄)), for all documents 📄?

Regular Document Spanners/Splitters

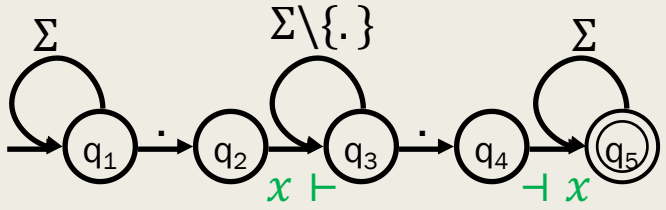
Information Extractors: Regular document spanners

Finite state automata with variable operations.
Essentially, regular expressions with capture variables, closed under relational algebra ($\cup, \pi, \bowtie$).

Splitters: Regular document Splitters

Unary regular document spanners

E.g. sentence extractor



Splitter Disjointness

Splitter Disjointness

A splitter is disjoint, if every part of a document is in at most one output document.

Disjoint Splitter

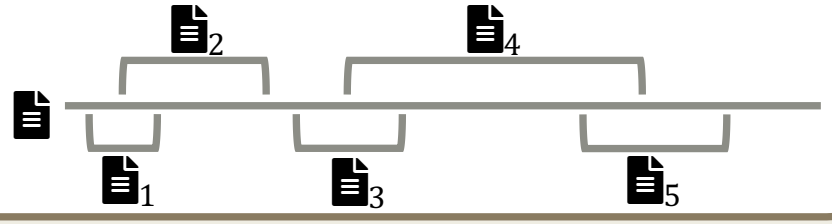
e.g.

- Sentence segmentation
- Split into paragraphs

Non-Disjoint Splitter

e.g.

- N-Grams
- Pairs of consecutive sentences



Main Results

Self-Splittability

Input: Information Extractor →, Splitter ✂
Question: Is →(📄) = →(✂(📄)), for all documents 📄?

Split-Correctness

Input: Information Extractors $\rightarrow_{global}$, $\rightarrow_{local}$, Splitter ✂
Question: Is $\rightarrow_{global}$(📄) = $\rightarrow_{local}$(✂(📄)), for all documents 📄?

Theorem 5.1 and Theorem 5.16

Self-Splittability and Split-Correctness for regular document spanners are PSPACE-complete

Theorem 5.7 and Theorem 5.17

Self-Splittability and Split-Correctness for disjoint splitters and regular document spanners in normal form* is in polynomial time

* deterministic + functional

Main Results

Splittability

Input: Information Extractor $\rightarrow_{global}$, Splitter ✂

Question: Is there an Information Extractor $\rightarrow_{local}$, such that $\rightarrow_{global}(📄) = \rightarrow_{local}(✂(📄))$ for any 📄?

Theorem 5.15

Splittability for disjoint splitters and regular document spanners is PSPACE-complete

Split-Constrained Black Boxes

- Information Extraction in practice is often done with other algorithms, e.g.
  - *coreference resolvers* ⟹ Splittable by paragraphs
  - *sentiment extractors* ⟹ Splittable by paragraphs or sentences
  - *named entity recognition* ⟹ Splittable by sentences or 5-grams

Black Box Split-Correctness

Input: Black Box Extractors $\rightarrow_1$, $\rightarrow_2$, a Splitter ✂, and a set C of Split-Constraints
Question: Is $\rightarrow_1 \bowtie \rightarrow_2$ splittable by ✂ under C?

Split-Constrained Black Boxes

Black Box Split-Correctness

Input: Black Box Extractors $\rightarrow_1$, $\rightarrow_2$, a Splitter ✂, and a set C of Split-Constraints
Question: Is $\rightarrow_1 \bowtie \rightarrow_2$ splittable by ✂ under C?

There are cases, where $\rightarrow_1 \bowtie \rightarrow_2$ is splittable by sentences, even though $\rightarrow_1$ is not!
All 'lost' tuples of $\rightarrow_1$ are not needed for the join

Theorem 7.3

There are extractors $\rightarrow_1$ and $\rightarrow_2$ that are self-splittable by the same splitter ✂
but $\rightarrow_1 \bowtie \rightarrow_2$ is not splittable by ✂ under C

Why should I care about this?

Main Motivation

Parallelized information extraction for huge documents

Further Motivation

1. Reduce the skew, if the a big number of smaller documents must be analyzed
2. Debugging: notify users if extractors reach beyond natural boundaries
3. Handling updates to the input document

Future Work

- Complexity of splittability for non-disjoint splitters
- Try to better understand splittability of split-constrained black boxes
- Empirical analysis of the framework

Thank you for your attention ☺

Paper: doi.org/10.1145/3294052.3319684