Counting Types for Massive JSON Datasets

BDA 2017, Nancy (présenté à DBPL 2017)

Mohamed-Amine Baazizi, Dario Colazzo, Giorgio Ghelli, Carlo Sartiani

Counting types

- Can types count?
- Should they?

Type theory perspective

- How to efficiently summarize the structure of large JSON datasets?
- How precise is the summary?

Database perspective

The first problem

- Type inference for massive JSON datasets, BDA 2016/ EDBT 2017
- We infer this type from a collection of JSON objects

```
{       title :  Str ;
        text : [ Str ] + Null  ;
        author : { address:T? ; affil:T? ;… } ?
        abstract : Str ?
}
```

- How «optional» is the author?
- How frequently a text is Null?

Let us count

$\{$ title : Str^{1000} ;

text : $([\ Str^{8000}\ ]^{800} + Null^{200})^{1000}$;

author : $\{$ add:T^{300} ; affil:T^{300} ;… $\}^{800}$

abstract : Str^{20}

$\}^{1000}$

The second problem

- How to capture correlation information?

- { **addr:T**^{300}; **aff:T**^{300}; r:T^{800} }^{800} **Concision**
- { **addr:T**^{300}; **aff:T**^{300}; r:T^{300} }^{300} + {r:T^{500} }^{500}
- { **addr:T**^{300}; r:T^{300} }^{300} + { **aff:T**^{300}; r:T^{500} }^{500}
- { **addr:T**^{300}; r:T^{500} }^{500} + { **aff:T**^{300}; r:T^{300} }^{300}
- { **addr:T**^{300}; r:T^{300} }^{300} + { **aff:T**^{300}; r:T^{300} }^{300} + {r:T^{200}}^{200} **Precision**

The type system

- $B ::= Null^i \mid Num^i \mid Str^i \mid Bool^i$
- $R ::= \{ l : T, \ldots, l : T \}^i$
- $A ::= [T]^i$
- $S ::= B \mid R \mid A$
- $T ::= S \mid 0 \mid T + T$
- Examples
  - Num^2 captures any multiset of two numbers
  - $[Num^4]^3$ a possible type for the multiset $\{ [1], [1], [1,2] \}^M$

The type inference algorithm

- Singleton
  - $\vdash \quad V : S$ $\qquad\qquad \vdash 3 : Int^1$
- Multiset
  - $\vdash v_1,\ldots,v_n :^M T$ $\qquad \vdash [1], [1], [1,2] :^M [\ Num^4]^3$
- Different abstraction levels
  - $[1], [1], [1,2] :^M [\ Num^4]^3$ **Concision**
  - $[1], [1], [1,2] :^M [\ Num^2]^2 + [\ Num^2]^1$
  - $[1], [1], [1,2] :^M [\ Num^1]^1 + [\ Num^1]^1 + [\ Num^2]^1$ **Precision**

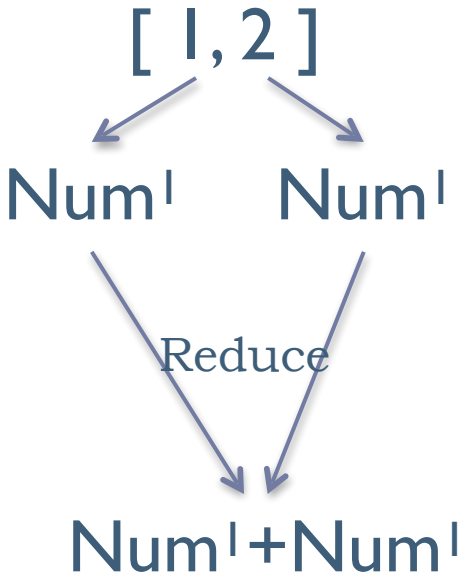
The type inference algorithm

- Singleton
  - $\vdash^{E}$ V : S
  - **Parametric inference**
  - $\vdash 3 : \mathrm{Int}^{1}$
- Multiset
  - $\vdash^{E} v_1,\ldots,v_n :^{M} T$
  - $\vdash [1], [1], [1,2] :^{M} [\ \mathrm{Num}^{4}]^{3}$
- Different abstraction levels
  - $[1], [1], [1,2] :^{M} [\ \mathrm{Num}^{4}]^{3}$ — **Concision**
  - $[1], [1], [1,2] :^{M} [\ \mathrm{Num}^{2}]^{2} + [\ \mathrm{Num}^{2}]^{1}$
  - $[1], [1], [1,2] :^{M} [\ \mathrm{Num}^{1}]^{1} + [\ \mathrm{Num}^{1}]^{1} + [\ \mathrm{Num}^{2}]^{1}$ — **Precision**

The type inference algorithm

$$\frac{\vdash^{E} \{\!|\, J_1, \ldots, J_n \,|\!\}^{\mathrm{m}} :^{\mathrm{m}} \mathbb{T}}{\vdash^{E} [J_1, \ldots, J_n] : [\mathbb{T}]^{1}} \quad (\textsc{TypeArray})$$

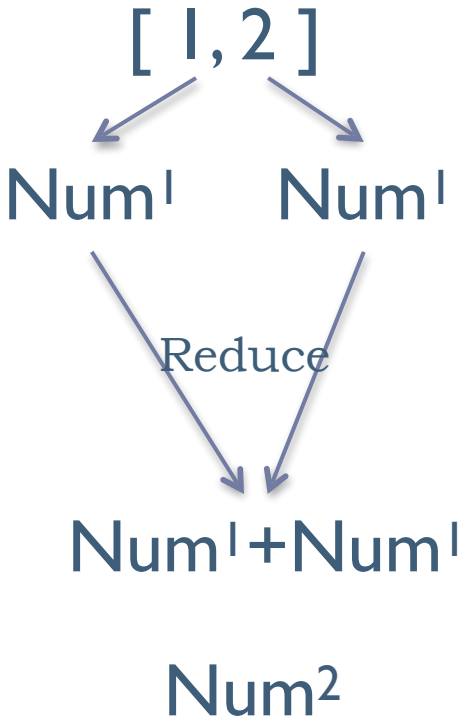
$$\frac{\vdash^{E} M_1 :^{\mathrm{m}} \mathbb{T}_1 \qquad \vdash^{E} M_2 :^{\mathrm{m}} \mathbb{T}_2}{\vdash^{E} M_1 \cup^{\mathrm{m}} M_2 :^{\mathrm{m}} \mathit{Reduce}(\mathbb{T}_1, \mathbb{T}_2, E)} \quad (\textsc{TypeUnionMultiset})$$



The type inference algorithm

$$\frac{\vdash^{E} \{\!|\, J_1, \ldots, J_n \,|\!\}^{\mathrm{m}} :^{\mathrm{m}} \mathbb{T}}{\vdash^{E} [J_1, \ldots, J_n] : [\mathbb{T}]^{1}} \quad (\textsc{TypeArray})$$

$$\frac{\vdash^{E} M_1 :^{\mathrm{m}} \mathbb{T}_1 \qquad \vdash^{E} M_2 :^{\mathrm{m}} \mathbb{T}_2}{\vdash^{E} M_1 \cup^{\mathrm{m}} M_2 :^{\mathrm{m}} \mathit{Reduce}(\mathbb{T}_1, \mathbb{T}_2, E)} \quad (\textsc{TypeUnionMultiset})$$



Parametric reduction

$$Reduce(\mathbb{T}_1, \mathbb{T}_2, E) =$$
$$\oplus(\quad \{\!| Merge(\mathbb{S}_1, \mathbb{S}_2, E) \mid \mathbb{S}_1 \in \circ\mathbb{T}_1, \mathbb{S}_2 \in \circ\mathbb{T}_2, E(\mathbb{S}_1, \mathbb{S}_2) |\!\}^{m}$$
$$\cup^{m} \{\!| \mathbb{S}_1 \mid \mathbb{S}_1 \in \circ\mathbb{T}_1, \nexists \mathbb{S}_2 \in \circ\mathbb{T}_2.\ E(\mathbb{S}_1, \mathbb{S}_2) |\!\}^{m}$$
$$\cup^{m} \{\!| \mathbb{S}_2 \mid \mathbb{S}_2 \in \circ\mathbb{T}_2, \nexists \mathbb{S}_1 \in \circ\mathbb{T}_1.\ E(\mathbb{S}_1, \mathbb{S}_2) |\!\}^{m} \quad)$$

$$Merge(\mathbb{B}^m, \mathbb{B}^n, E) = \mathbb{B}^{m+n}$$

$$Merge(\mathbb{R}_1, \mathbb{R}_2, E) =$$
$$\underline{\{} \ \{\!| (l, Reduce(\mathbb{T}_1, \mathbb{T}_2, E)) \mid (l, \mathbb{T}_1) \in \diamond(\mathbb{R}_1), (l, \mathbb{T}_2) \in \diamond(\mathbb{R}_2) |\!\}$$
$$\cup \{\!| (l, \mathbb{T}_1) \mid (l, \mathbb{T}_1) \in \diamond(\mathbb{R}_1), \nexists \mathbb{T}_2.\ (l, \mathbb{T}_2) \in \diamond(\mathbb{R}_2) |\!\}$$
$$\cup \{\!| (l, \mathbb{T}_2) \mid (l, \mathbb{T}_2) \in \diamond(\mathbb{R}_2), \nexists \mathbb{T}_1.\ (l, \mathbb{T}_1) \in \diamond(\mathbb{R}_2) |\!\}$$
$$\underline{\}}^{\#(\mathbb{R}_1)+\#(\mathbb{R}_2)}$$

$$Merge([\mathbb{T}_1]^m, [\mathbb{T}_2]^n, E) = [\ Reduce(\mathbb{T}_1, \mathbb{T}_2, E)\ ]^{m+n}$$

Equivalences of practical use

- Kind

$$\mathcal{K}(\mathbb{S}_1, \mathbb{S}_2) \Leftrightarrow kind(\mathbb{S}_1) = kind(\mathbb{S}_2)$$

$$kind(\mathsf{Null}^i) = 0 \quad kind(\mathsf{Bool}^i) = 1 \quad kind(\mathsf{Num}^i) = 2$$
$$kind(\mathsf{Str}^i) = 3 \quad kind(\mathbb{R}) = 4 \quad kind(\mathbb{A}) = 5$$

$\{$ **addr:T^{300}**; **aff:T^{300}**; r:T^{800} $\}^{800}$

- Label

$$\begin{aligned} \mathcal{L}([\mathbb{T}_1], [\mathbb{T}_2]) &\quad \text{always} \\ \mathcal{L}([\mathbb{B}], [\mathbb{B}]) &\quad \text{always} \\ \mathcal{L}(\mathbb{R}_1, \mathbb{R}_2) &\Leftrightarrow \text{Keys}(\mathbb{R}_1) = \text{Keys}(\mathbb{R}_2) \end{aligned}$$

$\{$ **addr:T^{300}**; r:T^{300} $\}^{300}$ + $\{$ **aff:T^{300}**; r:T^{300} $\}^{300}$ + $\{$r:T$^{200}\}^{200}$

Kind reduction: twitter data

$$\{\ \text{contributors: } (\text{Null}^{9{,}599{,}980} + [\text{Num}^{20}]^{20})^{9{,}600{,}000}\ ;$$

$$\text{retweeted} : \text{Bool}^{9{,}600{,}000}\ ;$$

$$\text{retweeted\_status}\ \{\ldots\} : \{\ldots\}^{1{,}200{,}000}\ ;$$

$$\text{deleted} : \{\ldots\}^{300{,}000}\ ;$$

$$\}^{9{,}900{,}000}$$

Label reduction: twitter data

$\{ \text{con}: \ldots^{7,200,000}; \text{ret}: \text{Bool}^{7,200,000}; \ldots\}^{7,200,000}$

$+\{ \text{con}: \ldots^{1,200,000}; \text{ret}: \text{Bool}^{1,200,000}; \ldots\}^{1,200,000}$

$+\{ \text{con}: \ldots^{1,040,000}; \text{ret}: \text{Bool}^{1,040,000}; \text{r\_s}: \{\}^{1,040,000}; \ldots\}^{1,040,000}$

$+\{ \text{con}: \ldots^{160,000}; \text{ret}: \text{Bool}^{160,000}; \text{r\_s}: \{\}^{160,000}; \ldots\}^{160,000}$

$+\{ \text{deleted}: \{\}^{300,000}; \ldots\}^{300,000} :$

Label reduction: twitter data

$\{ \text{con}: \ldots^{7,200,000}; \text{ret}: \text{Bool}^{7,200,000}; \ldots\}^{7,200,000}$

$+\{ \text{con}: \ldots^{1,200,000}; \text{ret}: \text{Bool}^{1,200,000}; \ldots\}^{1,200,000}$

$+\{ \text{con}: \ldots^{1,040,000}; \text{ret}: \text{Bool}^{1,040,000}; \text{r\_s}: \{\}^{1,040,000}; \ldots\}^{1,040,000}$

$+\{ \text{con}: \ldots^{160,000}; \text{ret}: \text{Bool}^{160,000}; \text{r\_s}: \{\}^{160,000}; \ldots\}^{160,000}$

$+\{ \text{deleted}: \{\}^{300,000}; \ldots\}^{300,000} :$

$$\{ \text{contributors}: (\text{Null}^{9,599,980} + [\text{Num}^{20}]^{20})^{9,600,000} ;$$
$$\text{retweeted} : \text{Bool}^{9,600,000} ;$$
$$\text{retweeted\_status} \{\ldots\} : \{\ldots\}^{1,200,000} ;$$
$$\text{deleted} : \{\ldots\}^{300,000} ;$$
$$\}^{9,900,000}$$

Kind reduction

Experiments

- Scala implementation
- Spark cluster of 5+1 nodes, 64GB, 100 cores
- 3 real life datasets :
  - Github (1m objects /10GB / 14 sec)
  - Twitter (9.9m objects / 21GB / 53 sec)
  - Nytimes (1.2m objects / 21GB / 27 sec)
- Extraction of interesting features

Related work

- PL community: dependent types, probabilistic types
- Dataguide avec statistiques [Klettke et al. 2016]
- JavaScript library for MongoDB [Schmit 2017]
- Approaches w/o counting information
- No parametric approach, so far

[Klettke et al. 2016] Schema Extraction and Structural Outlier Detection for JSON-based NoSQL Data Stores, Technologie und Web (BTW)
[Schmidt. 2017]. mongodb-schema. (2017). https://github.com/mongodb-js/mongodb-schema.

To sum up

- An algorithm to summarize JSON data:
  - Well defined semantics
  - Parametric
  - Parallel
  - Yielding quantitative information
- What else may a counting type do?

Thank you!