Linguistic Tools for Managing Grammatical Domains

Anders Miltner
Devon Loehr
Arnold Mong
Kathleen Fisher
David Walker
Let’s Talk About Dates

Date my_date = new Date("05/26/22");
Bad Dates

https://en.wikipedia.org/wiki/Time_formatting_and_storage_bugs
Don't forget to attend LangSec on 26/05/22!
(26 May 2022)

Don't forget to attend LangSec on 26/05/22!
(2026 May 22)
This can be addressed by creating a formal parser!

But there's so many possible parsers we might want to create!
Grammar Induction
Learn a Grammar from some form of specification

- L* Algorithm [Angluin, 1987]
  - Seminal, but exponential in worst case
- Can we do better?
  - Not unless we break RSA! [Angluin and Kharitonov, 1995]
  - SOTA algorithms work via biasing to grammars they think occur in practice
Our Approach

Let users encode the set of possible grammars and the biases in grammar generation.
Workflow for Grammar Induction

Metagrammar describing the grammatical domain

Set of Positive Examples

Grammar within the grammatical domain

Set of Negative Examples
Benefits

Few Examples Required

Explicitly Encoded Biases

Generated Grammar Guarantees
Metagrammar by Example

```
1 Sep -> ? "," named COMMA
2   ? "/" named SLASH
3   ? ";" named DASH.
4 constraint(|Production(Sep)| = 1)

6 Digit -> ["0"-"9"].

8 Year -> ? Digit Digit
9   ? Digit Digit Digit Digit.
10 constraint(|Productions(Year)| = 1)

12 Month -> ? Digit
13   ? "0" Digit
14       | "10" | "11" | "12".
15 constraint(|Production(Month)| = 2)

17 Day -> ? ["1" - "9"]
18   ? "0" ["1" - "9"]
19       | ["1" - "2"] Digit | "30" | "31".
20 constraint(|Production(Day)| = 2).

22 Date -> ? Day Sep Month Sep Year
23   ? Month Sep Day Sep Year
24   ? Year Sep Month Sep Day
25   ? Year Sep Day Sep Month.
26 constraint(|Production(Date)| = 1).

28 preference prefer SLASH 2.0.
29 preference prefer DASH 1.0.

31 start Date
```
Work-In-Progress

• Improving efficiency of grammar induction algorithm
• Improving speed of parsing
• Improving surface language with better syntactic sugar
Questions

• By focusing on individual domains, grammar induction becomes easier

• Metagrammar files encode possible grammars and grammatical biases for selection

• Future work in algorithmic improvements and improvements to the surface language