

Parsing, Constraints und Suche

Arne Köhn

13. Januar 2015

Frobbing

- ▶ Suchalgorithmus für CDG
- ▶ Transformationen: 1-2 Kanten durch andere ersetzen
- ▶ Constraintgesteuert

Data: sentence S

Result: Analysis of S

```

1 List removedConflicts  $\leftarrow$  [ ];
2 Analysis best  $\leftarrow$  makeInitialAnalysis(S);
3 Analysis current  $\leftarrow$  best;
  // we need to know from which conflict we have started
4 Conflict initialConflict  $\leftarrow$  getHardestSolvableConflict(current);
5 while solvable conflicts remain do
6   Conflict c  $\leftarrow$  getHardestSolvableConflict(current);
7   current  $\leftarrow$  attackConflict(c, current, removedConflicts);
8   if score(current) > score(best) then
9     best  $\leftarrow$  current;
10    reset(); // resets internal state of attackConflict
    // store with which conflict we'll start the next iteration
11    initialConflict  $\leftarrow$  getHardestSolvableConflict(current);
12  else if current == null then
13    setUnresolvable(initialConflict);
14    current  $\leftarrow$  best;
15    removedConflicts  $\leftarrow$  [ ];
16  end
17 end
18 return best;
```

Algorithm 1: frobbing

Data: Conflict c , Analysis $current$, List $removedConflicts$

Result: Analysis not violating c

```

1 removedConflicts += c;
2 patches ← generatePatches(c);
3 bestPatch ← null;
4 foreach patch in patches do
5     patch.judge();
6     if detectCircle(patch, current, removedConflicts) then
7         continue; // ignore patches that reintroduce conflicts
8     end
9     if patch > bestPatch then
10        | bestPatch ← patch;
11    end
12 end
13 if bestPatch == null then
14     return null;
15 end
16 Analysis newA ← bestPatch.apply(current);
17 return newA;

```

Algorithm 2: attackConflict

Constraints und Suche in Frobbing

- ▶ Constraint satisfaction bei Frobbing
- ▶ Suche bei Frobbing
 - ▶ Suchraumgröße?
 - ▶ Zyklenerkennung?
 - ▶ Art der Suche?

RBGParser

- ▶ Komplexe Bewertungskriterien
- ▶ Frobbing: Fehlergesteuerte Suche
- ▶ Was tun, wenn Fehler nicht lokal sind?
- ▶ Oder einfach alles mehr oder weniger schlecht bewertet wird?

RBGParser (Cont.)

- ▶ Sei χ Die Menge aller möglichen Bäume für unseren Satz
- ▶ Gegeben: Bewertungsfunktion $f(x) : \chi \times \mathbb{R}$

Gesucht: $\operatorname{argmax}_b f(b)$, wie findet man es?

Steps to Excellence

- ▶ Nutze Sampling um nach einer guten Lösung zu suchen
- ▶ Gibbs sampling: $N-1$ Variablen festhalten, eine sampeln
 - ▶ Oft benutzt, wenn Wahrscheinlichkeiten nicht direkt berechnet werden können
 - ▶ Super für Bayes-Netze
 - ▶ Generiert gemeinsame Verteilung aus den bedingten
 - ▶ Nachteile?

Steps to Excellence (Cont.)

- ▶ Mehrere (z.B. 4) gleichzeitig sampeln!
- ▶ Vom Wort Kopf sampeln, bis wir am Baum angekommen sind.
- ▶ \Rightarrow Blocked Gibbs Sampling
- ▶ Bäm! Es Funktioniert! Aber wie genau?

Woher Wahrscheinlichkeit für Sampling nehmen?

- ▶ $f(x)$ geht nicht — Warum?

Woher Wahrscheinlichkeit für Sampling nehmen?

- ▶ $f(x)$ geht nicht — Warum?
- ▶ Wir müssten alle möglichen Samples betrachten
 - ▶ Wenn N Kanten in Satz Länge L gesamlet werden

Woher Wahrscheinlichkeit für Sampling nehmen?

- ▶ $f(x)$ geht nicht — Warum?
- ▶ Wir müssten alle möglichen Samples betrachten
 - ▶ Wenn N Kanten in Satz Länge L gesampelt werden
 - ▶ L^N Möglichkeiten!
 - ▶ z.B. $20^4 = 160.000$ Möglichkeiten!
- ▶ Stattdessen: nur $P(\text{Kante})$ nutzen; später $f(b)$ berechnen

Optimale Splits für Untertitel

- ▶ Gegeben: Text und Alignment zu Audio
- ▶ Aufgabe: Optimal Text in Untertitel aufteilen
- ▶ Bewertungsfunktionen:
 - ▶ Bestrafe falsche Länge
 - ▶ Bestrafe Splits zwischen zwei Wörtern
- ▶ Wie findet man idealen Split?