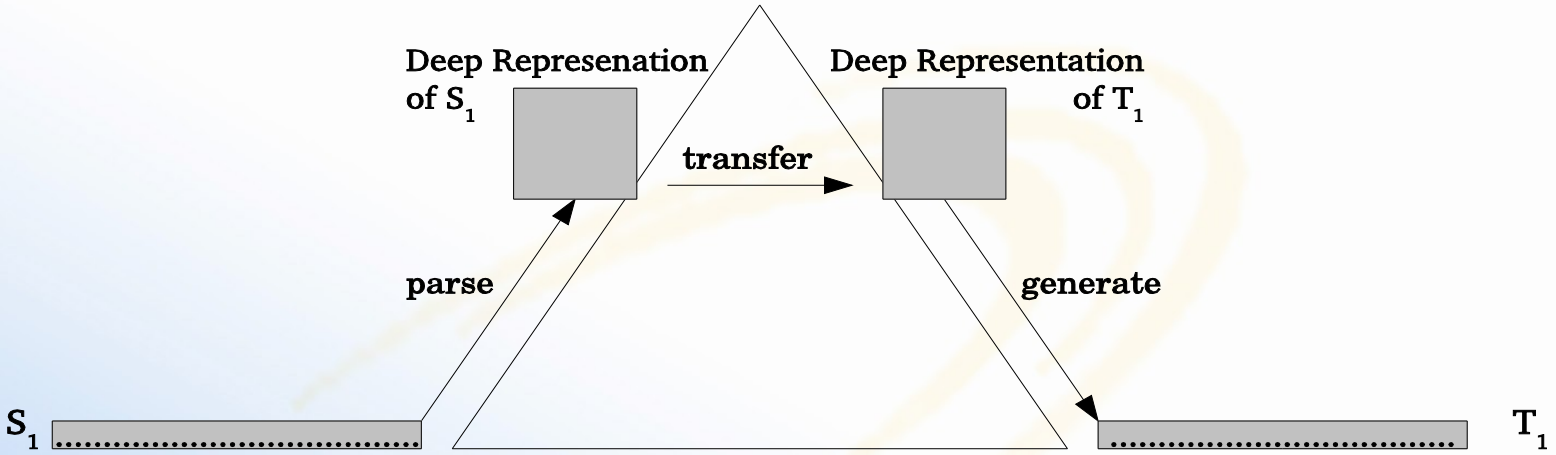
Probabilitistic Transfer-based MT

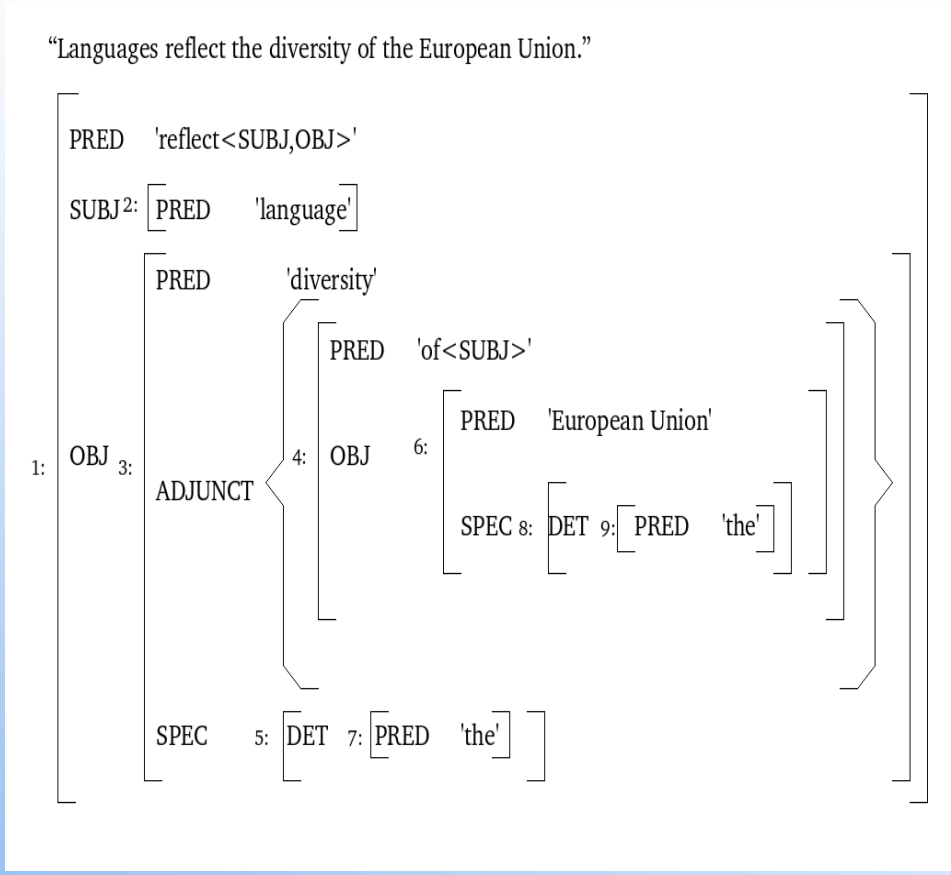
Yvette Graham, Josef van Genabith
National Centre for Language Technology
School of Computing
Dublin City University

Transfer-based Machine Translation



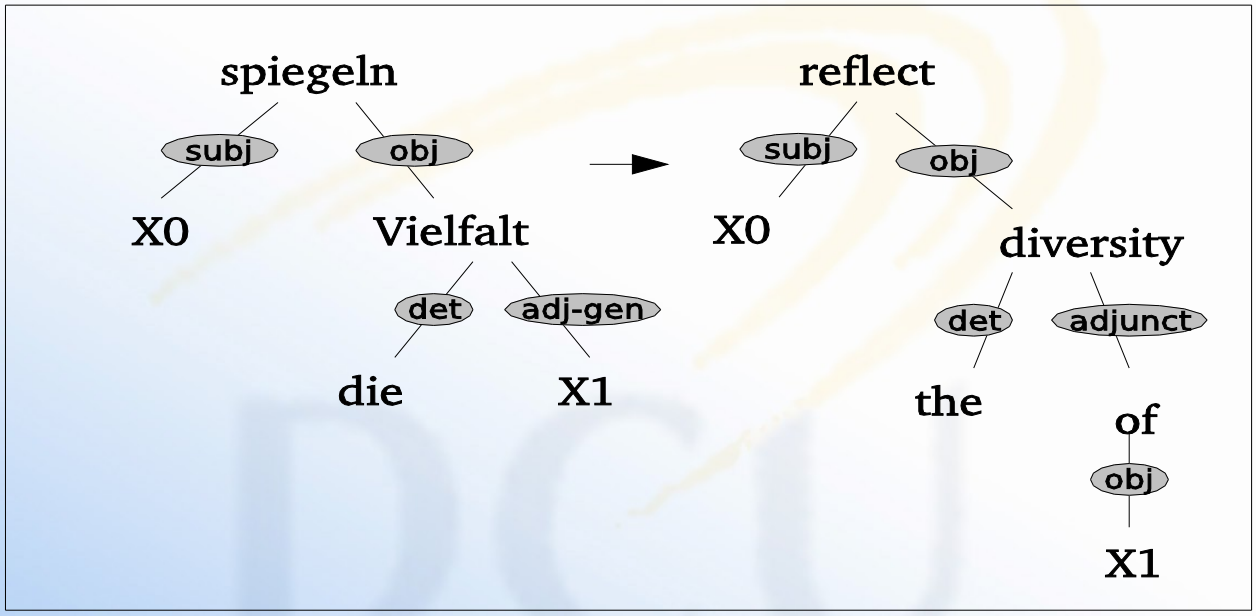
- Transfer-rule Induction Algorithm & Packed Rule Data Structure
- Probabilistic Decoder for Transfer-based MT

Why LFG F-structure?



- encode **dependencies** between the content words of a sentence
- encode the **linguistic information** necessary to generate the TL surface form sentence
- provide an abstract linguistic structure for forming **linguistically motivated generalizations** about how to translate well from one language to another
- abstracting away from surface form to f-structure produces **more general rules** that can be applied to multiple surface form examples – reducing data sparseness problems
- reliable and accurate linguistic **resources** for parsing and generation are vital to the overall success of a transfer-based system

Example F-structure Transfer Rules



X0 speigel(t) die Vielfalt **X1** wider. → **X0** reflect(s) the diversity of **X1**.

Transfer Rule Induction

- Exponential number of possible transfer rules within a single f-structure pair

- Lots of those rules are complete rubbish!

The Plan: Constrain rule induction to eliminate unwanted rules

Constraining Rule Induction

- Contiguity Constraint (Riezler & Maxwell 06)

- Cross-structural Consistency Constraint:

  Use a 1-1 set of alignments between nodes & allow any SL-TL node pair with equivalent non-empty sets of *aligned descendents* be a transfer-rule root.

Aligned Descendent of a SL local f-structure n:

Any descendent of n that is aligned with a TL f-structure.

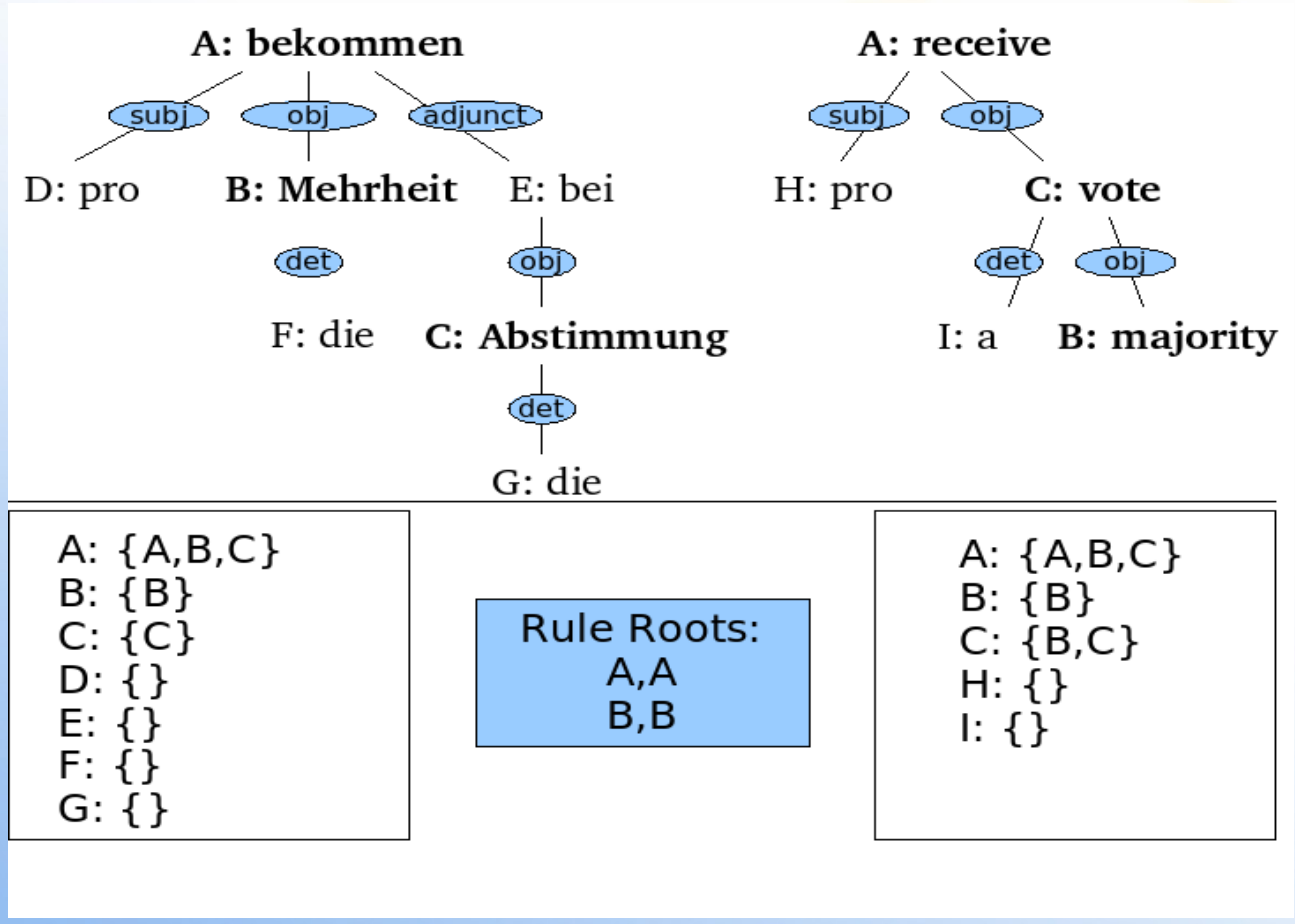
If n itself is aligned with any TL local f-structure, then n is considered an aligned descendent of itself.

Note: by this definition the actual root nodes of every f-structure training pair is a rule root.

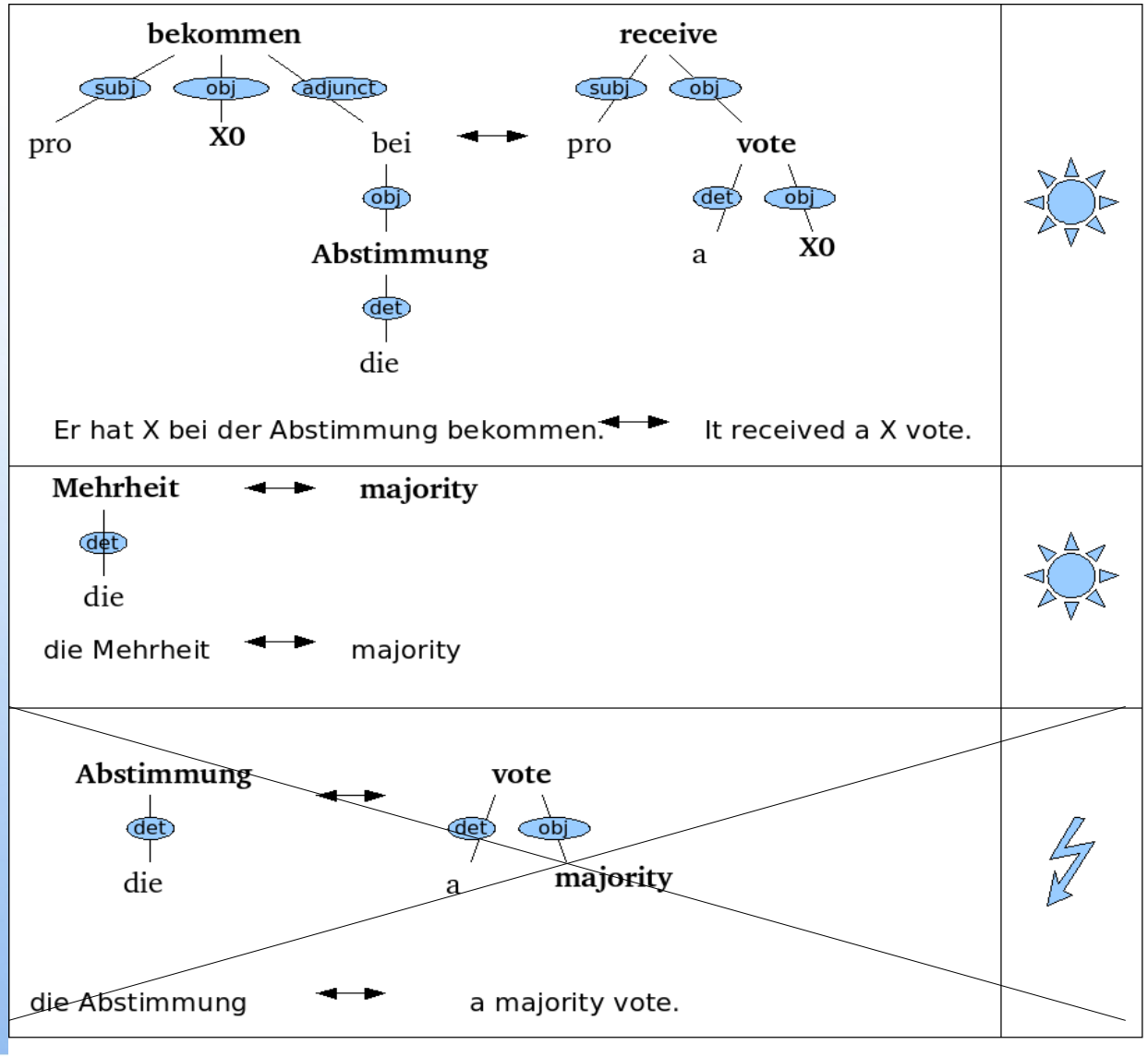
Example

Er hat die Mehrheit bei der Abstimmung bekommen.

It received a majority vote.



Induced Rules



Putting Variables into Transfer Rules

Variable Constraints:

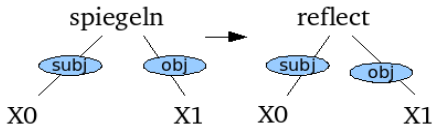
1. The root of a transfer rule may never be a variable.

2. For any non-root node:

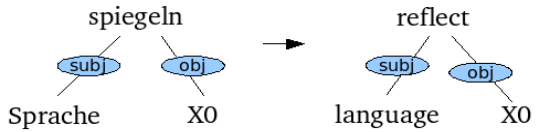
   - Iff it is aligned with a TL node it can be a variable

Example F-structure Transfer Rules

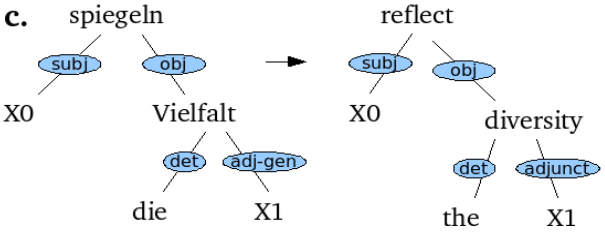
a.



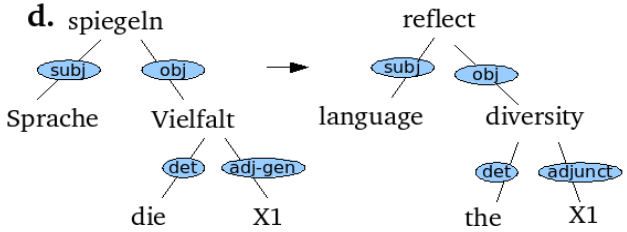
b.



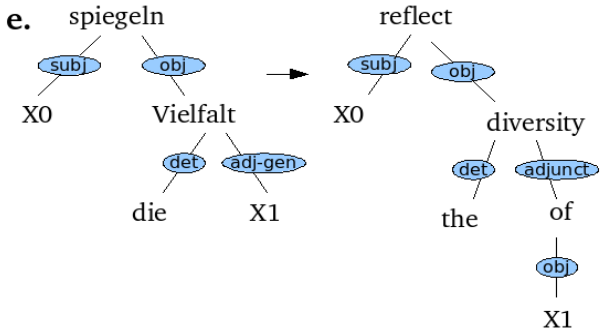
c.



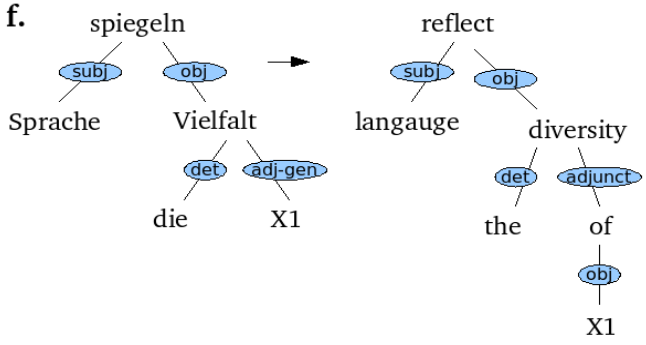
d.



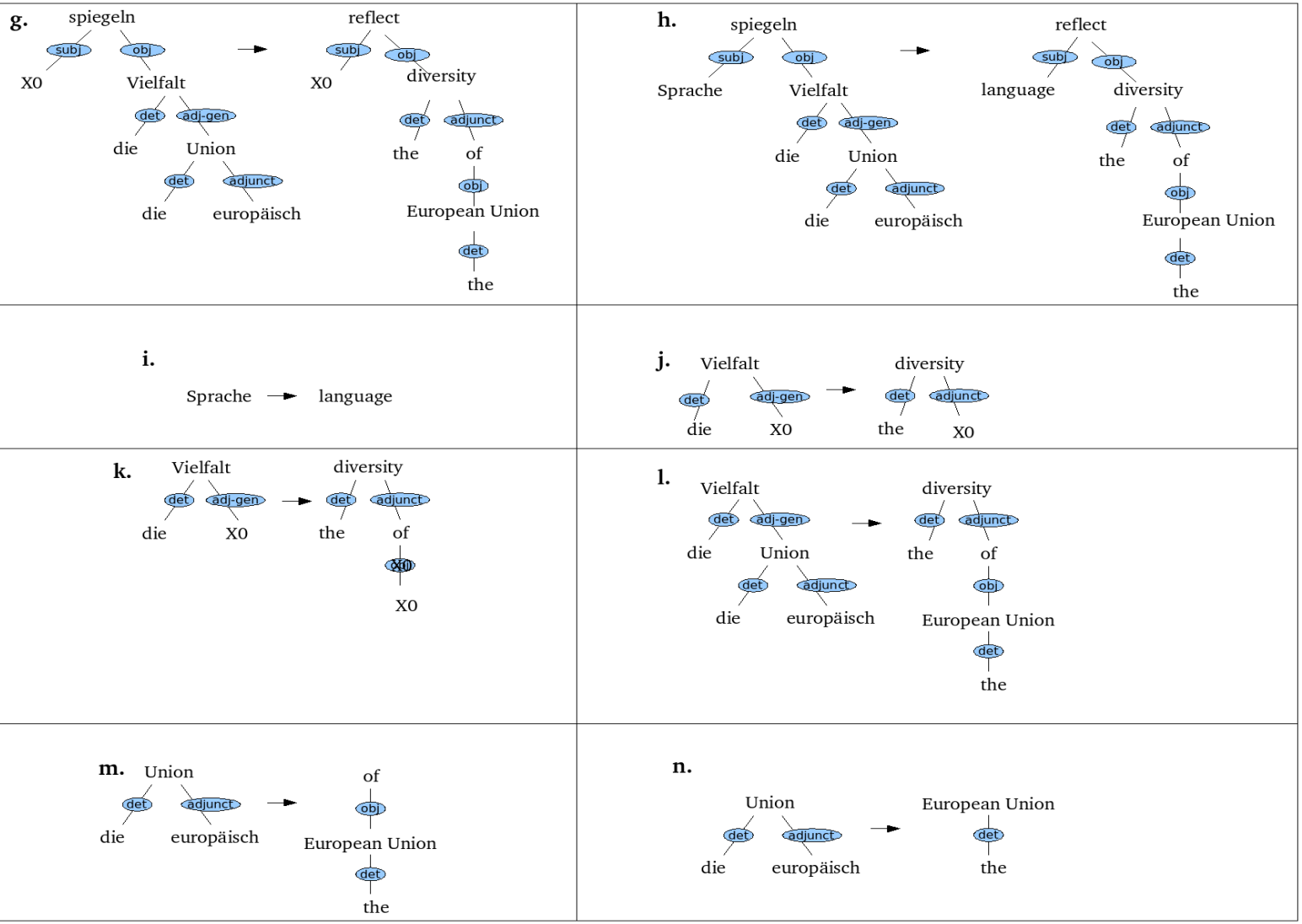
e.



f.



Example F-structure Transfer Rules



Packed Transfer Rule Data Structure

• Encode all possible rules that can be induced from a SL TL f-structure pair given these constraints in a single packed representation

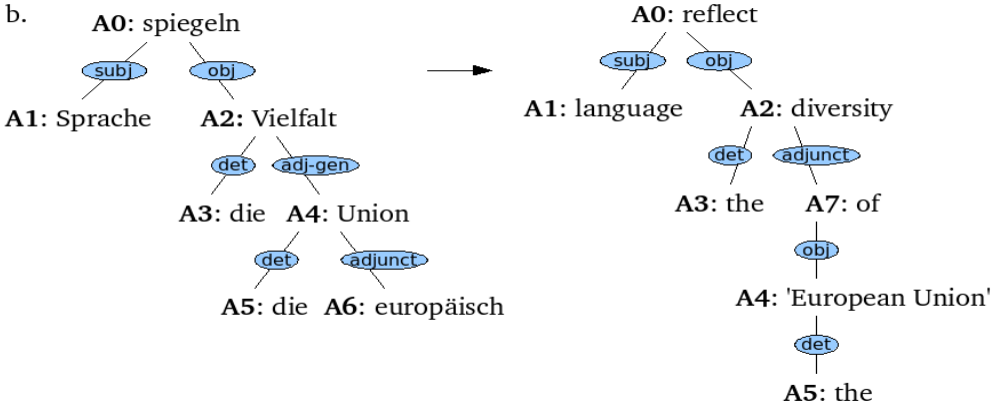
a.

pred(**A0**, X1, 'spiegeln')
subj(**A0**, X1, X2)
obj(**A0**, X1, X3)
topic(**A0**, X1, X2)
pred(**A1**, X2, 'Sprache')
pred(**A2**, X3, 'Vielfalt')
adj-gen(**A2**, X3, X4)
spec(**A2**, X3, X5)
det(**A2**, X5, X8),
pred(**A4**, X4, 'Union')
adjunct_member(**A4**, X4, X6)
spec(**A4**, X4, X7)
det(**A4**, X7, X9)
pred(**A6**, X6, 'europäisch')
subj(**A6**, X6, X4)
pred(**A5**, X9, 'die')
pred(**A3** , X8, 'die')

→

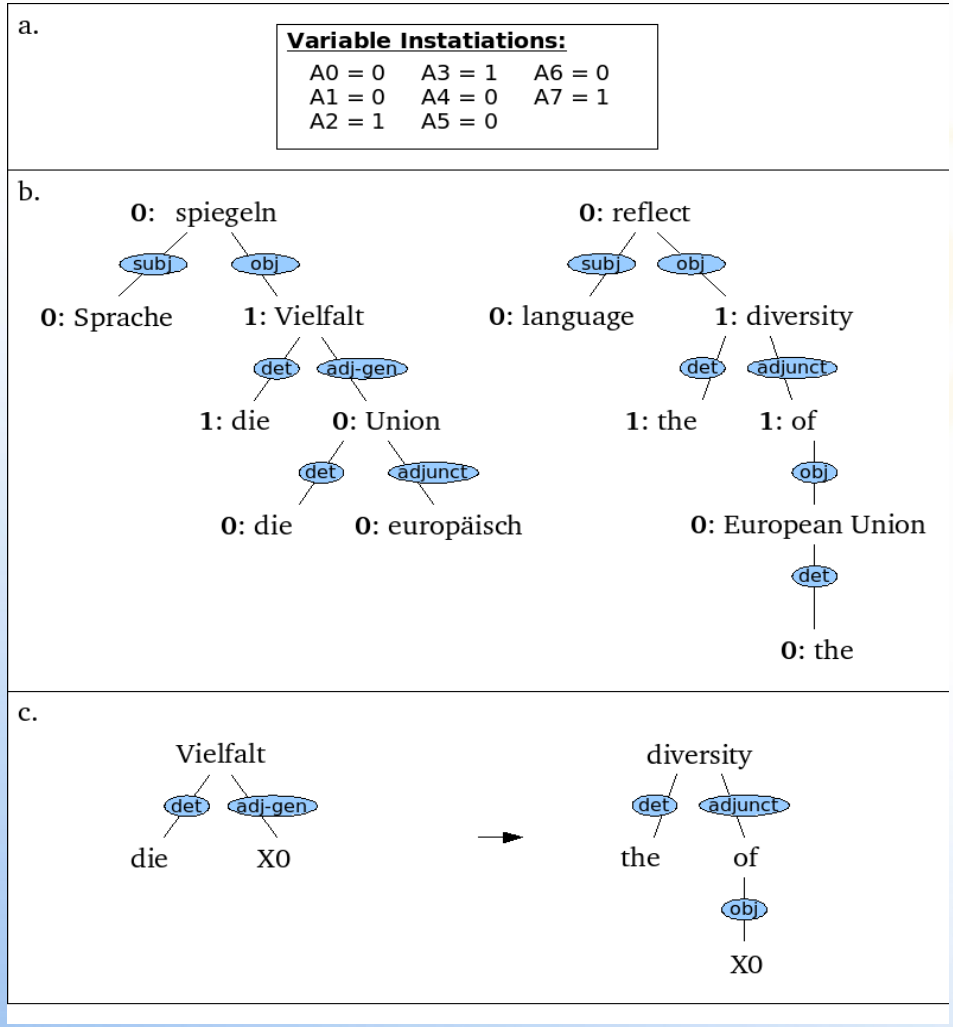
pred(**A0**, X1, 'reflect')
subj(**A0**, X1, X2)
obj(**A0**, X1, X3)
pred(**A1**, X2, 'language')
pred(**A2**, X3, 'diversity')
adjunct_member(**A2**, X3, X4)
spec(**A2**, X3, X5)
det(**A2**, X5, X7)
pred(**A7**, X4, of)
obj(**A7**, X4, X6)
pred(**A4**, X6, 'European Union')
spec(**A4**, X6, X8)
det(**A4**, X8, X9)
pred(**A5**, X9, 'the')
pred(**A3**, X7, 'the')

b.



Example Unpacking Rule

- Unpack a rule by assigning true or false to certain context variables



- To unpack all possible rules assign all possible combinations of true and false to the context variables

Experimental Evaluation of Packed Data Structure

- Extracted all rules from 217,666 f-structure pairs of length 5-15 of German-English Europarl
- Unpacked 10 sets of rules – each set consisting of rules induced from randomly selected sets of 1000 training pairs, on average 23955 rules per set

		Disk Space		Write Time		Load Time		Retrieving/Unpacking	
Set	**No. Rules**	**Enum**	**Packed**	**Enum**	**Packed**	**Enum**	**Packed**	**Enum**	**Packed**
1	24121	96.37M	7.2M	144s	128s	211s	17s	2s	3s
2	24486	98.89M	7.16M	145s	127s	215s	19s	2s	3s
3	23650	93.58M	7.17M	142s	133s	200s	18s	2s	2s
4	23882	96.83M	7.22M	149s	118s	210s	18s	2s	2s
5	24146	98.03M	7.15M	148s	128s	212s	17s	2s	3s
6	23355	91.75M	7.1M	140s	128s	198s	21s	2s	3s
7	23620	94.55M	7.21M	141s	142s	204s	18s	2s	2s
8	23687	94.02M	7.11M	137s	124s	201s	17s	1s	3s
9	23534	94.95M	7.12M	142s	120s	204s	17s	1s	3s
10	25069	100.66M	7.26M	152s	231s	219s	19s	2s	2s
Average	23955	95.96M	7.17M	144s	137.9s	207.4s	18.1s	1.8s	2.6s
All Rules Estimate	5214189	**20.4G**	**1.52G**	8h43m	8h20m	**12h32m**	**1h06m**	**6m31s**	**9m24s**

Chart-based Decoder

Input: a pre-compiled chart for the source language f-structure containing

Output: TL f-structure or n-best TL f-structures

- Top-down beam search of transfer chart
- Log-linear model to combine feature scores
- MERT to adjust weights (still to be integrated into decoder)

Future Work

- Fully integrate **MERT** into decoder
- Implement a **faster search algorithm** for decoding

- **Factor atomic features** and values to investigate what level of rule specificity is best
- Automatically **learn where atomic features get their values** – SL v TL