

Generation in MT



Progress Report #2

Jan Hajič

The Team

– Senior members & affiliate members

- Jan Hajič, Charles Univ., Prague
- Drago Radev, Univ. of Michigan
- Gerald Penn, Univ. of Toronto
- Jason Eisner, Johns Hopkins Univ.
- Owen Rambow, Univ. of Pennsylvania
- Dan Gildea, Univ. of Pennsylvania
- Bonnie Dorr, Univ. of Maryland

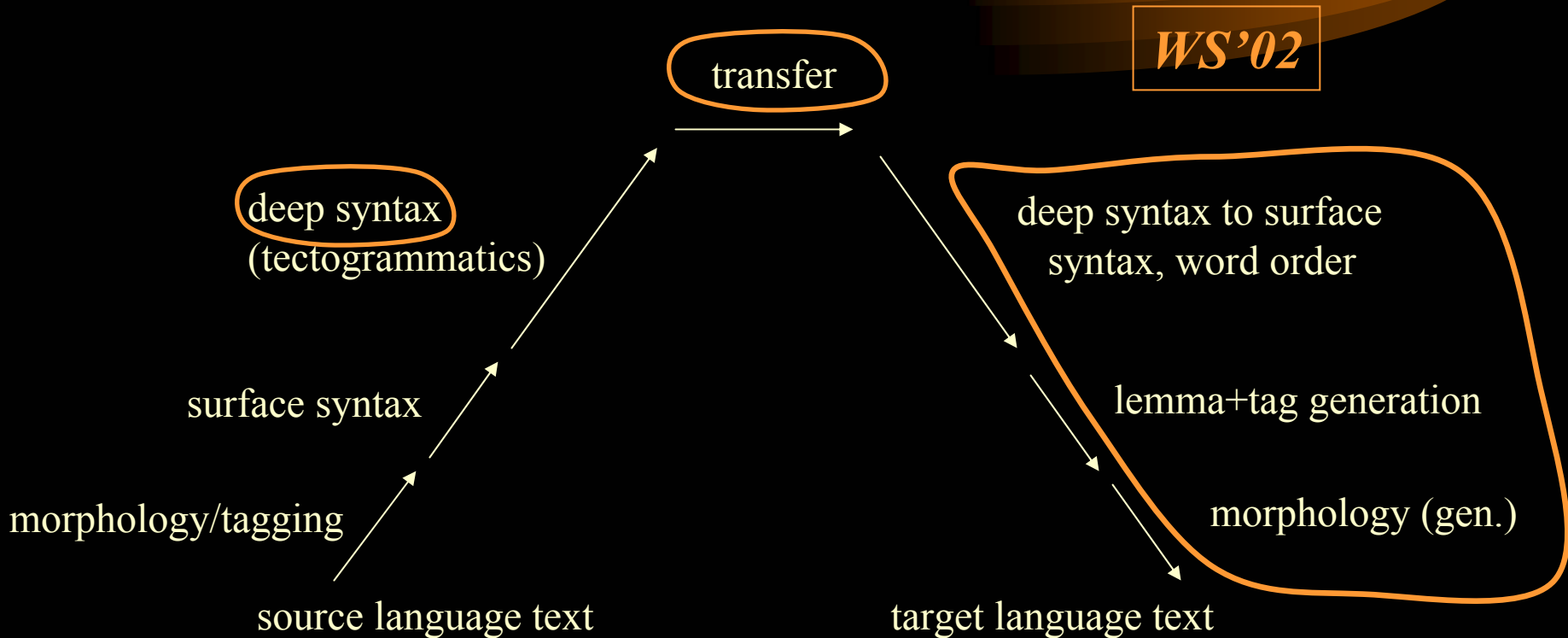
– Students:

- Yuan Ding, Univ. of Pennsylvania, Martin Čmejrek, Charles Univ., Prague
 - Terry Koo, MIT, Kristen Parton, Stanford Univ.
 - Jan Cuřín, Ivona Kučerová (Charles University)

The Goal

- Generate English (plain surface form)
 - from syntactic-semantic sentence representation (so-called “tectogrammatical”, or TR)
- Possible application setting:
 - machine translation
 - other uses:
 - part of front-end for QA systems, full generation
- Evaluate under various circumstances

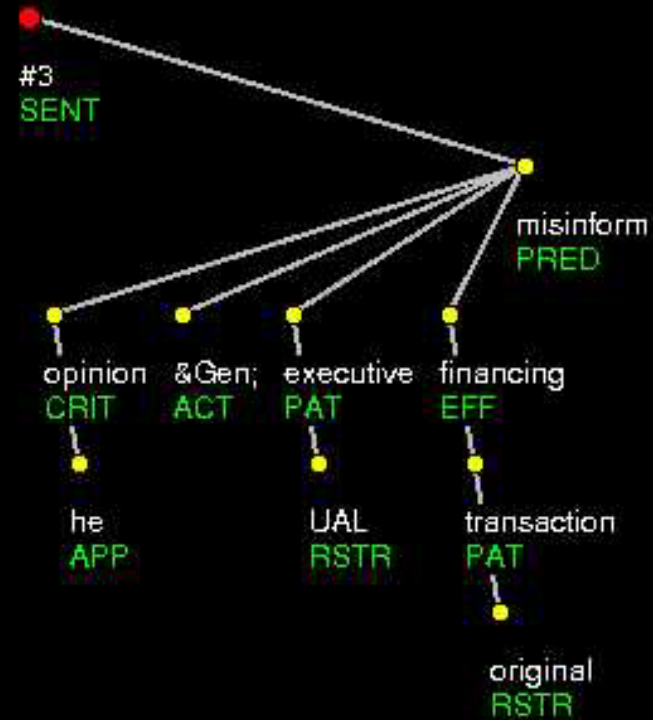
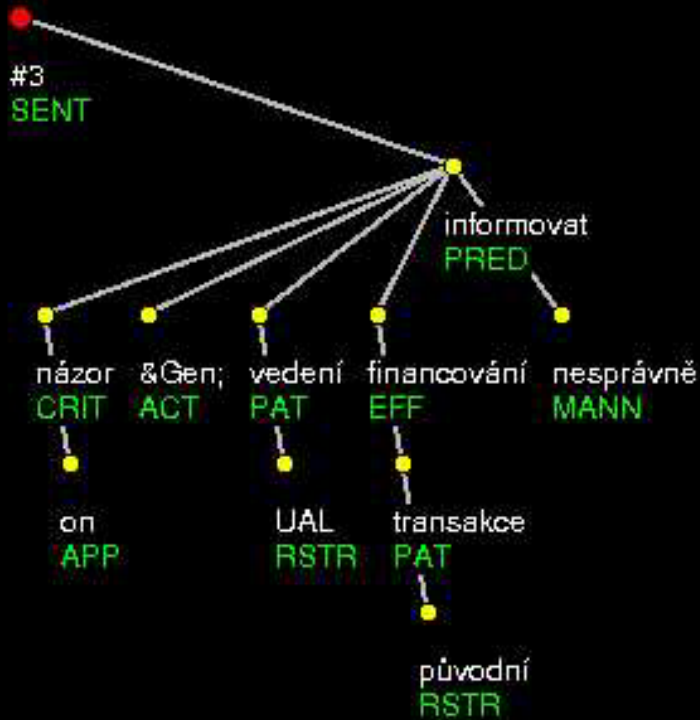
The Framework



Tectogrammatical representation

NULL Vedení UAL bylo podle jeho názoru o financování původní transakce nesprávně informováno.

3/51: #3 According to his opinion UAL 's executives were misinformed about the financing of the original transaction .



Final Datasets

- Data
 - Automatically parsed Czech, to TR
 - translation of WSJ, 11000 sentences
 - Penn treebank lemmatized & with lex./heads, AR
 - Penn Treebank at TR, automatically converted
 - Evaluation data (~250+250 sentences)
 - manually created English TR sentences
 - automatically created English TR sentences
 - Trivially translated Czech TR to English TR

Data Summary

	tr	de	te	sde	ste
auto AR, TR	42697	242	248	3384	1416
man TR	561	242	248	199	0
czech	7987	242	248	2942	1051
retransl	0	242	248	0	0

tr ... training

de ... dev test

te ... eval test

sde ... step dev test

ste ... step eval test

The pipeline



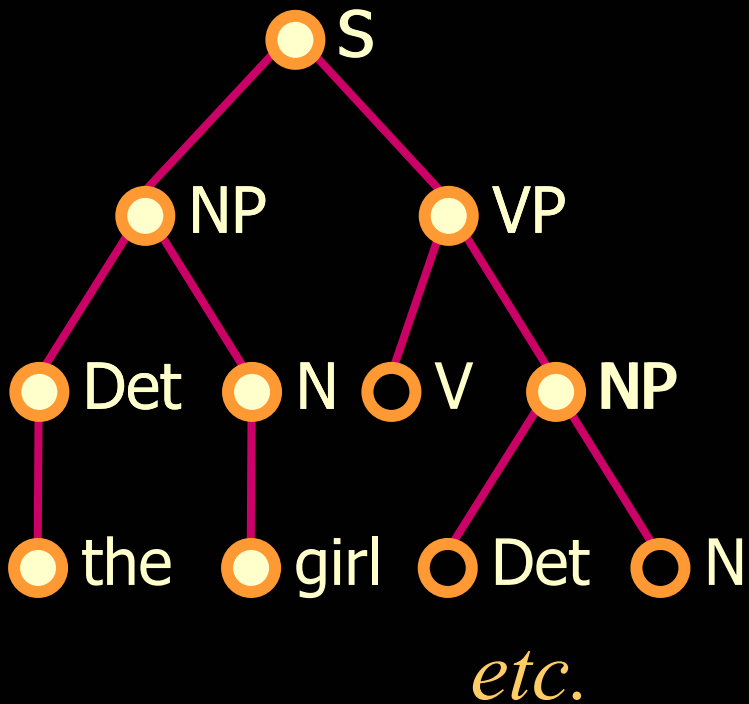
- Channel model
 - English TR to English AR
 - no morphology, no punctuation, no word order
 - Possibly classifiers “inside” (prepositions, ...)
- Word order
 - Collins’s model, on AR
- Punctuation insertion
- English morphology

Tree-to-tree mappings

- MT or generation turns old trees into new trees.
 - Or, noisy channel turns new trees into old trees.
- Our team has tree pairs. We want to infer process that turned one tree into the other.
- Assuming process is local, we need an alignment among tree nodes.
 - Not a 1-to-1 alignment. Also 1-0, 2-1, etc.
 - So use **tree substitution grammar**.

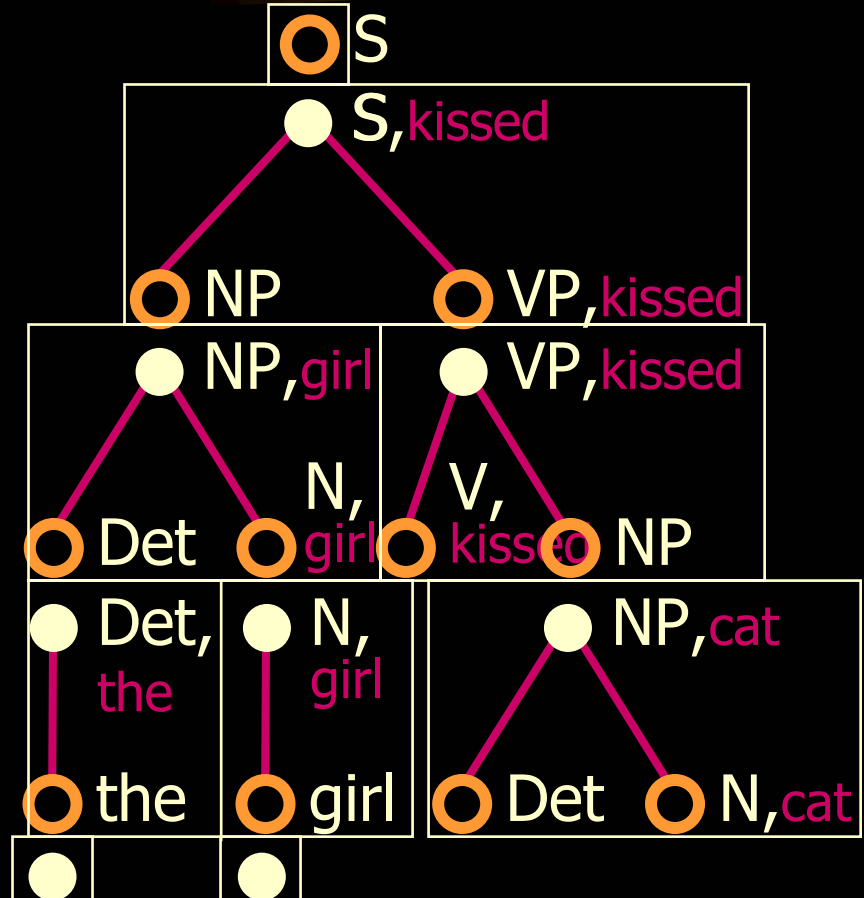
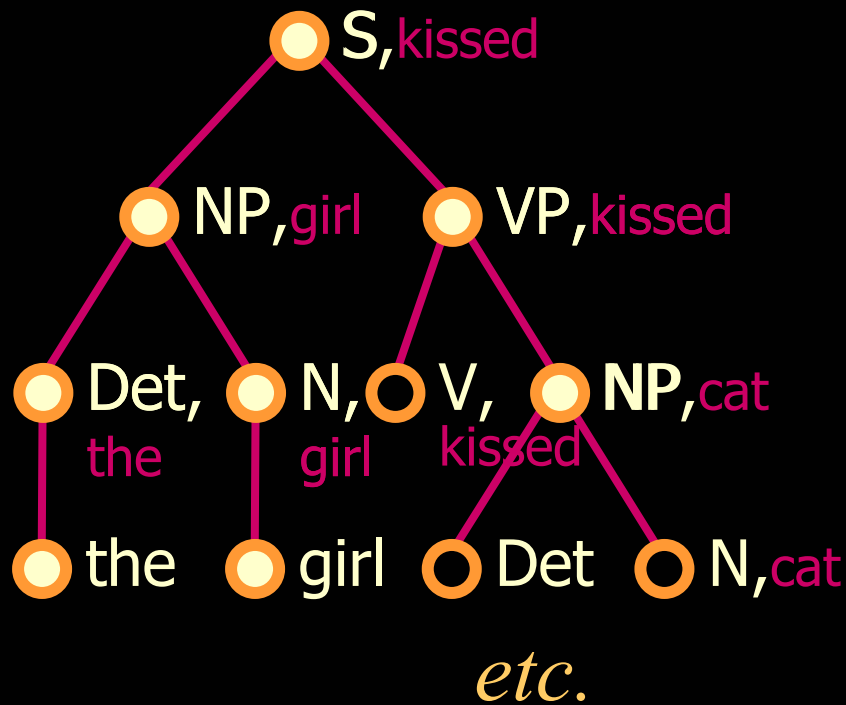
Context-Free Grammar

“the girl kissed her cat”



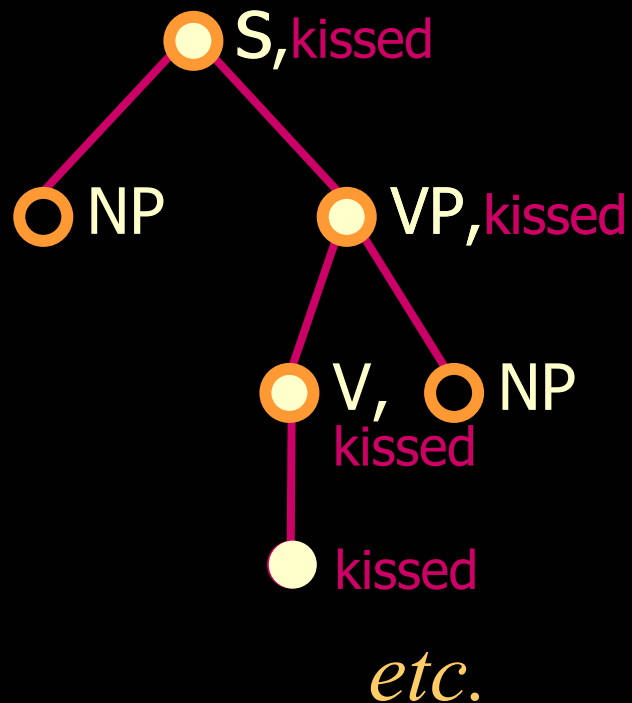
Lexicalized Context-Free Grammar

“the girl kissed her cat”



Lexicalized Context-Free Grammar

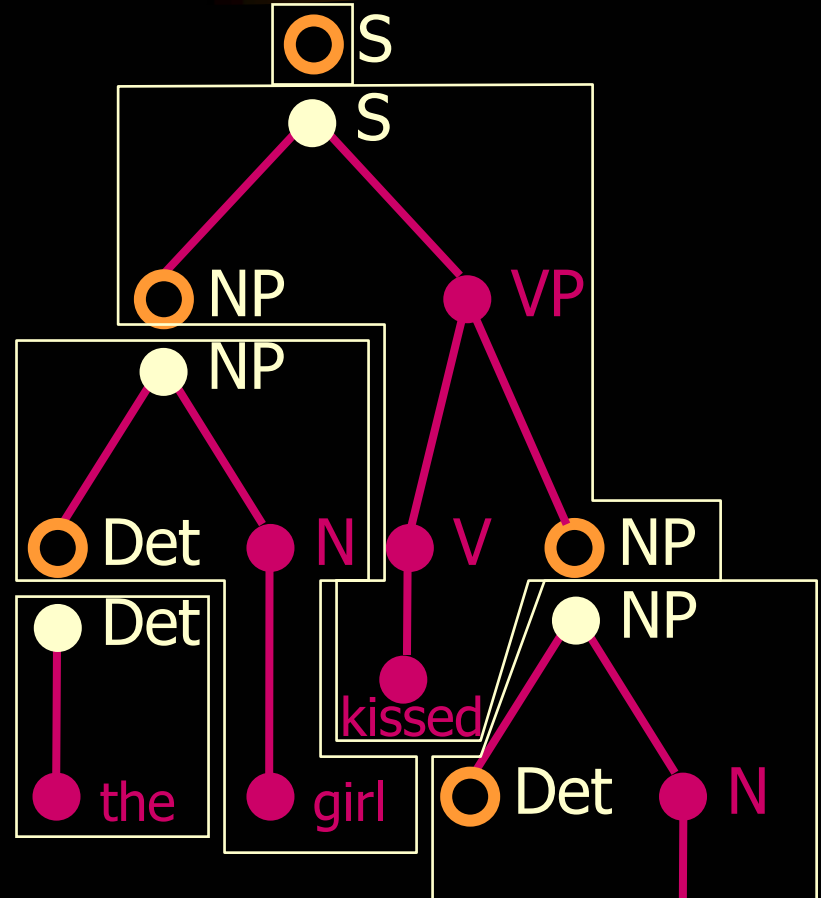
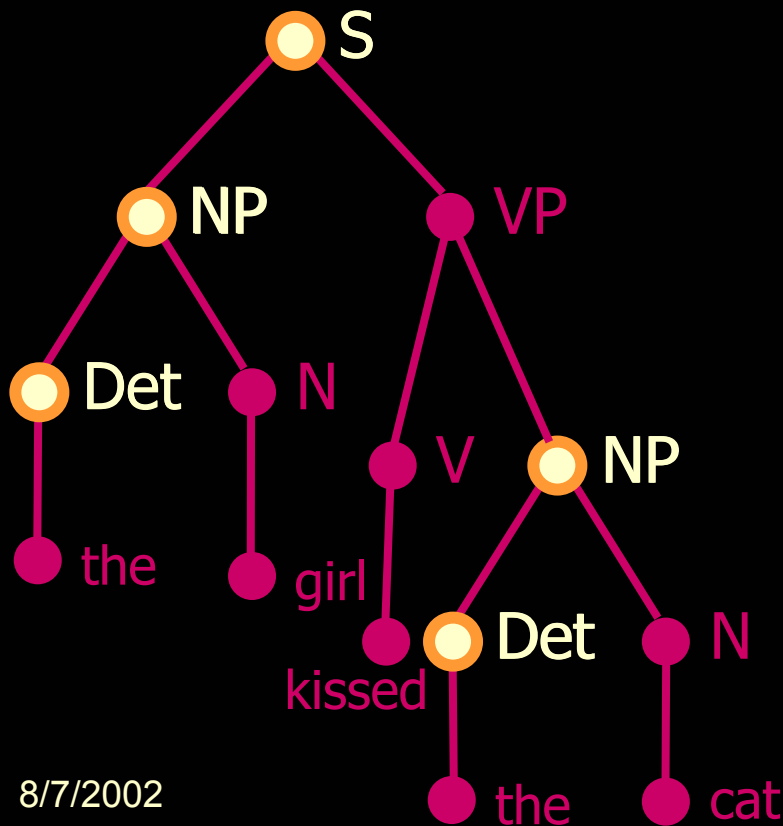
“the girl kissed her cat”



look at all the rules
headed by kissed ...

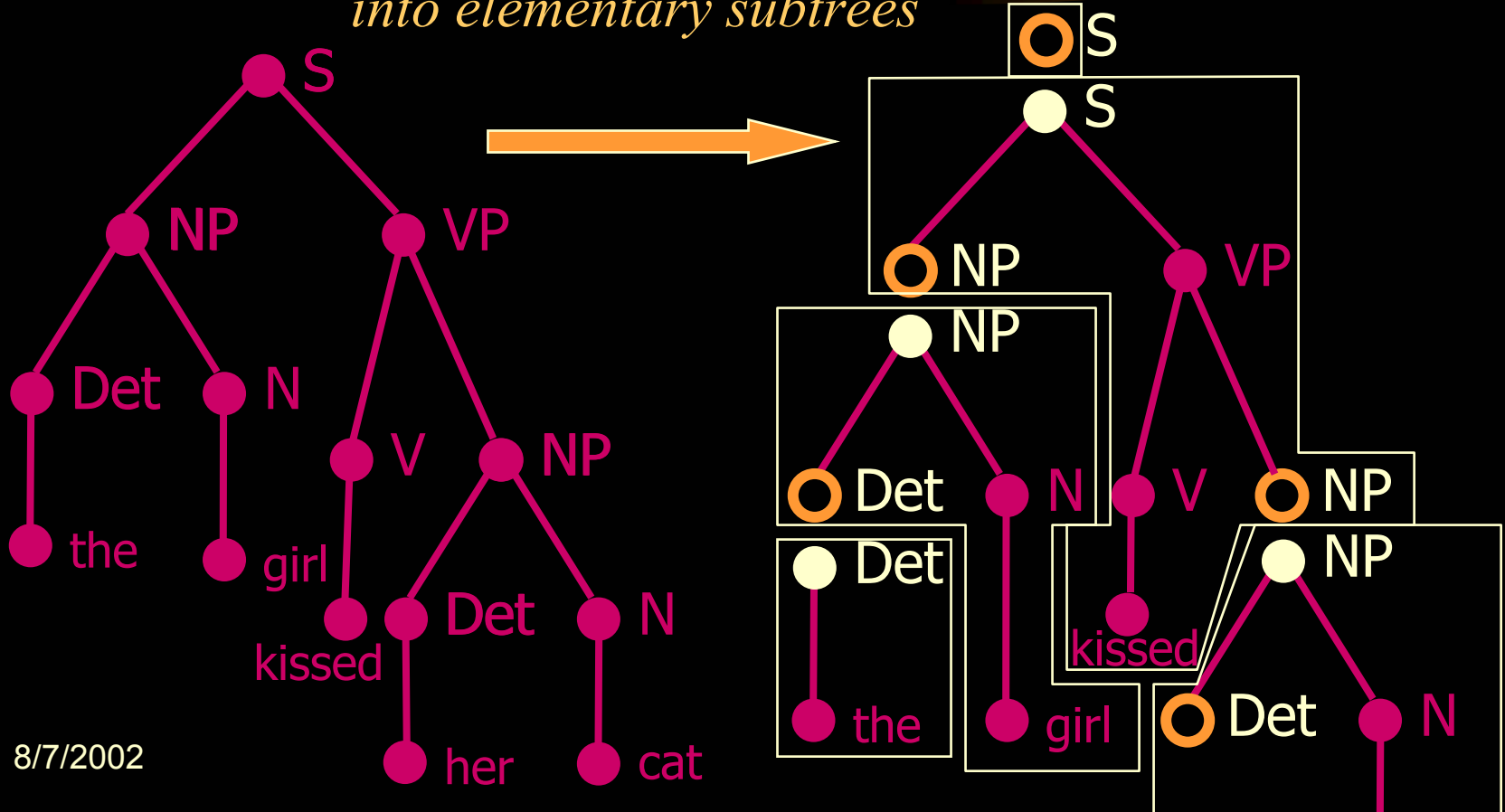
Lexicalized Tree Substitution Grammar

“the girl kissed her cat”



Lexicalized Tree Substitution Grammar

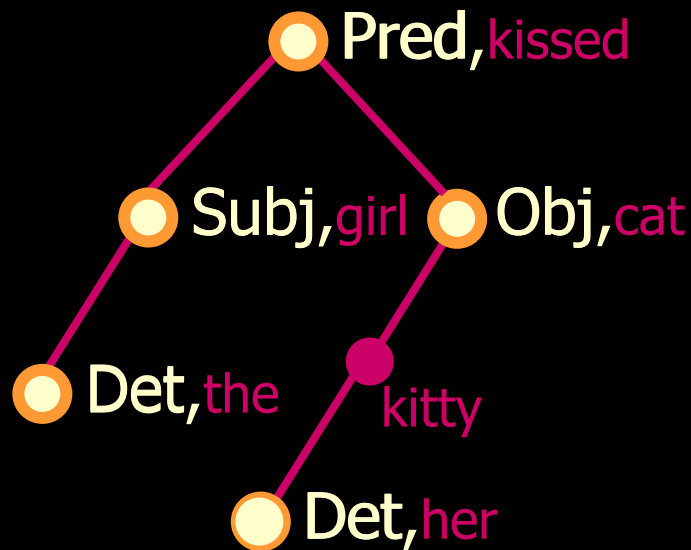
one "parse" of the tree
into elementary subtrees



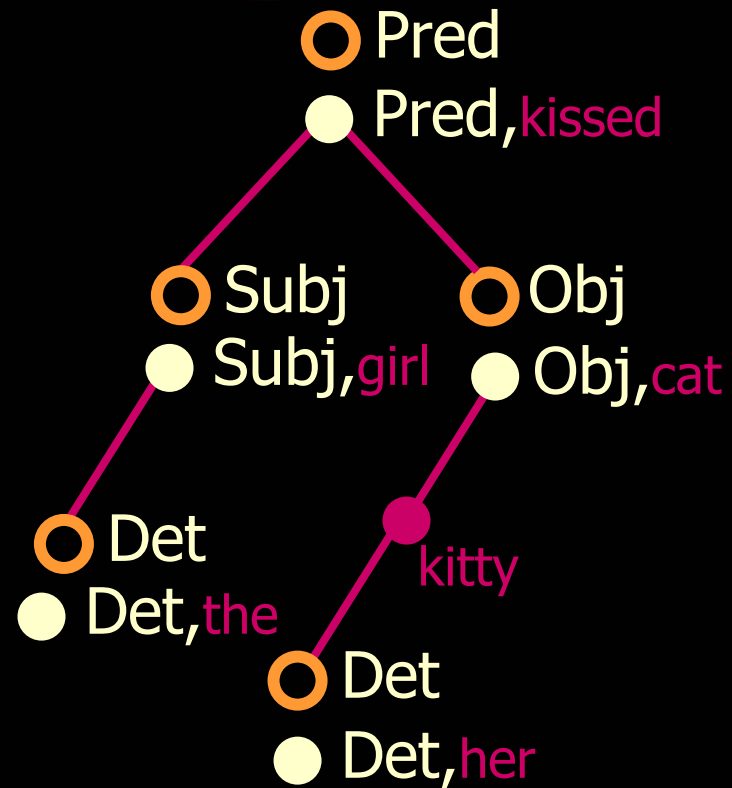
Dependency-Style

Lexicalized Tree Substitution Grammar

*“the girl kissed
her kitty cat”*



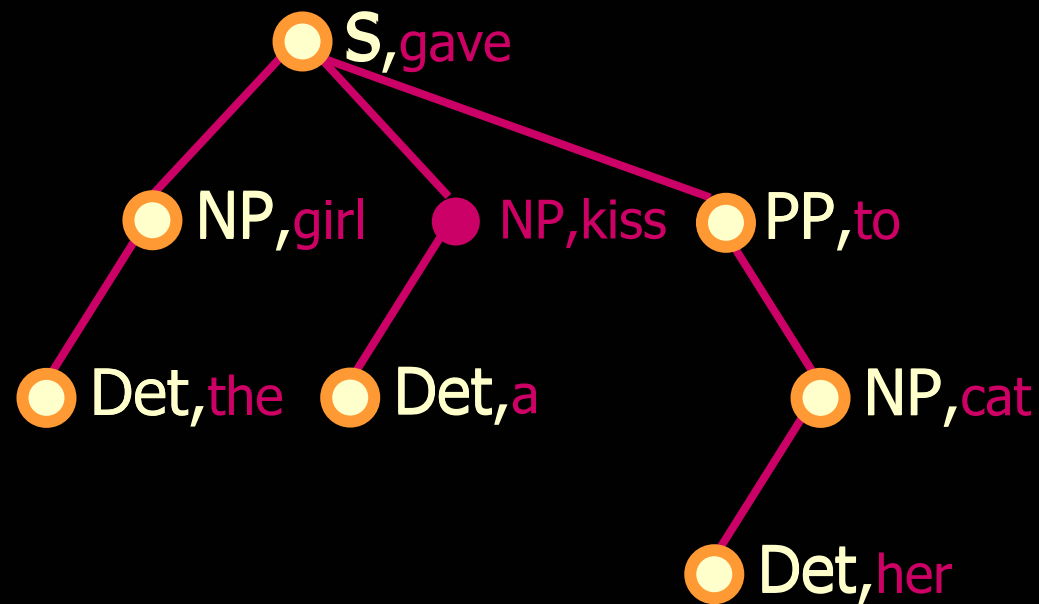
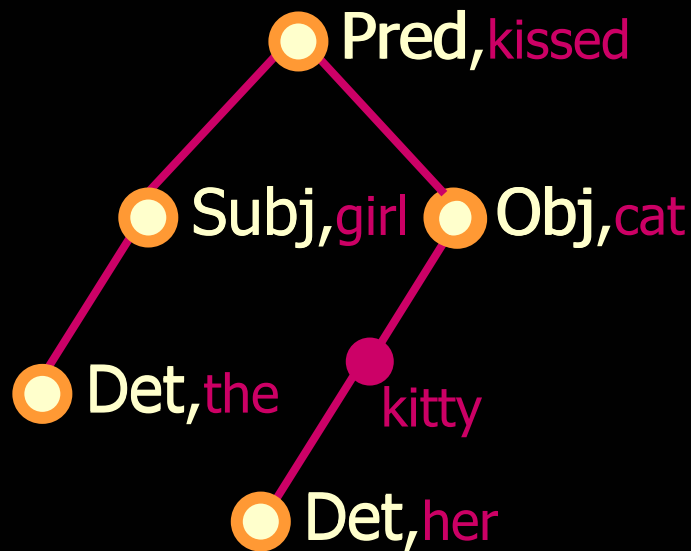
kiss(girl(the),cat(kitty(her)))



Synchronous Dependency-Style Lexicalized Tree Substitution Grammar

*“the girl kissed
her kitty cat”*

*“the girl gave a kiss
to her cat”*

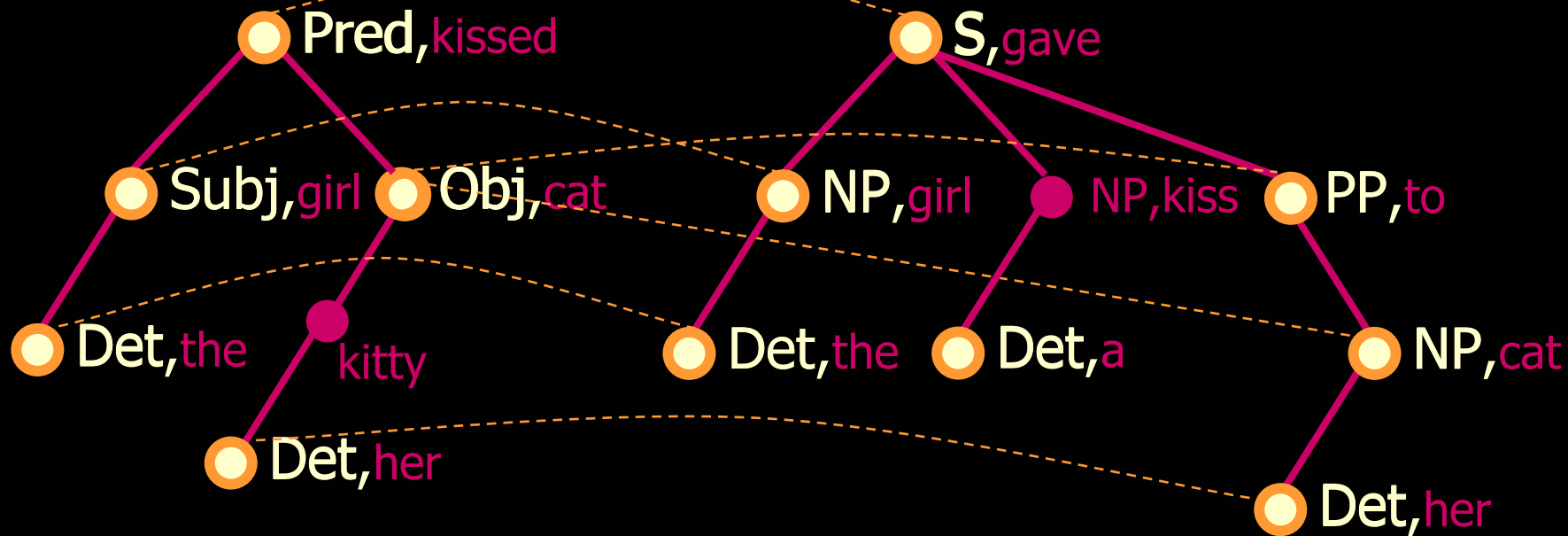


kiss(girl(the),cat(kitty(her)))

Synchronous Dependency-Style Lexicalized Tree Substitution Grammar

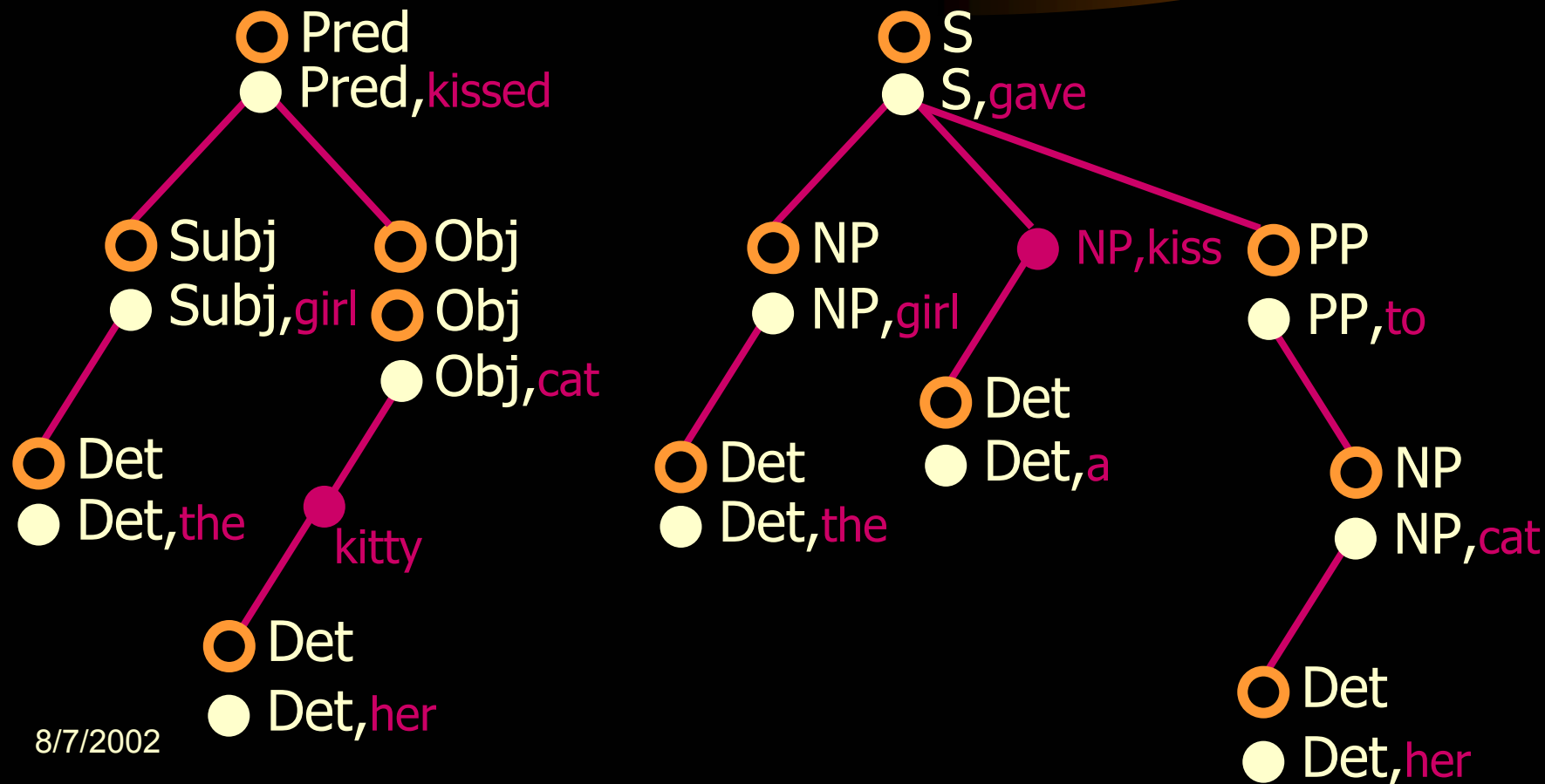
*“the girl kissed
her kitty cat”*

*“the girl gave a kiss
to her cat”*



Condition generation of t_1, t_2
on their joint root nonterminals

$$P(T_1, T_2) = \prod p(t_1, t_2 \mid n)$$



$$P(T1, T2) = \prod p(t1, t2 | n)$$

How This Simplifies Things

- Alignment: find A to max $P_{\theta}(T1, T2, A)$
 - Decoding: find T2, A to max $P_{\theta}(T1, T2, A)$
 - Training: find θ to max $\sum_A P_{\theta}(T1, T2, A)$
- } Noisy channel okay
- Do everything on little trees instead! $p_{\theta}(t1, t2, a)$
 - Align or decode possible little trees; stitch possibilities together by dynamic programming
 - Then retrain using EM counts

Penn TB + PropBank + LCS → TR

- **Goal:** use PropBank annotations to
 - Improve automatic construction of English TRs
 - Allow generation from “generic” pred-arg structures
- **Tasks**
 - Augment PropBank with roleset info ✓
 - Add lexical-conceptual role tags ✓
 - Convert to TG ✓
- **Results**
 - Not using PropBank: 53.7% error in functor tags
 - Using PropBank: 43.2% error (19.6% reduction)

Word order

- Experiments:
 - Tree-based models:
 - Analytical level surface dependency, tree-based
 - Collins model
 - Uses function information (Sb, Obj, Atr, ...)
 - 94% (chance: 68%)
 - levels ≥ 7 nodes ignored: 1.5% of nodes abs.
 - No punctuation (inserted later)

Punctuation

- C5.0 decisions tree classifier
- Trained on Eng AR data (sect. 0-20 WSJ)
 - with commas stripped
- Labelling:
 - NO-ACTION, INSERT-RIGHT
- Events:
 - features from local & global context
 - all from MSFT paper & more (sentence length, ...)
- Runs but no good results yets
 - lower baseline for WSJ than technical (36% vs. 52%)

Preposition insertion

- C4.5 decision tree classifier
- Trained on TR English (lemma/functor/POS)
 - with “aligned” prepositions
- Labelling: one label per preposition (“insert-to”)
- Events:
 - no lexicalization
 - lots of lexicalization (too slow training)
- Current status:
 - no lexicalization: precision/recall 66%/50% on insertions
 - running simpler lexicalization now

Hybrid approach

- FUF/Surge (Elhadad/Robin): almost everything needs to be specified
- Using a three-step process:
 - technically converting TR to LISP representation
 - converting to acceptable FUF input (tree-to-tree)
 - Running FUF
- Current situation
 - top-level works, node-level works -> integration

Evaluation

- Four tracks x <Channel model, FUF>
 - Track 1: from automatically generated Eng
 - Track 2: from manually created Eng
 - Track 3: from improved automatic (PropBank)
 - Track 4: from Czech TR (simple translation)
- Comparison to Stat. MT (Egypt, word-based)
- TR lemmas baseline
- Metric & software: BLEU