Detecting and Correcting Syntactic Errors in Machine Translation Using Feature-Based Lexicalized Tree Adjoining Grammars

Wei-Yun Ma

Kathleen McKeown

Columbia University

Motivation

- Statistical MT output is often ungrammatical because of the lack of sufficient linguistic knowledge for the target language

Examples

Many young student play basketball.

The boy play basketball and the girl also play basketball.

The boy play basketball and he asks the girl play basketball.

Examples Using MS Word

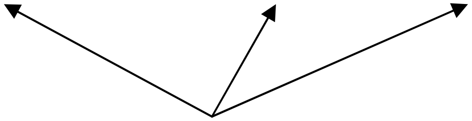
Many young student play basketball.

The boy play basketball and the girl also play basketball.

The boy play basketball and he asks the girl play basketball.

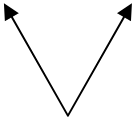
Our Goal of Detection

"Many young student play basketball."

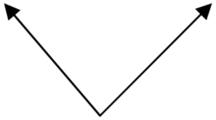


agreement error

"The boy play basketball and the girl also play basketball."

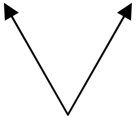


agreement error

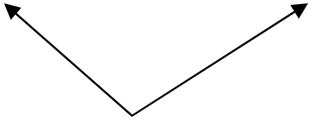


agreement error

"The boy play basketball and he asks the girl play basketball."



agreement error



mode error

What we need to achieve the goal

- A lexicalized grammar for target language
  - Natural and intuitive to define lexical item’s syntactic usage
  - Easy to maintain and easy to extend
- The ability to simultaneously detect multiple ungrammatical types and their corresponding words based on the grammar

What we need to achieve the goal

XTAG English Grammar

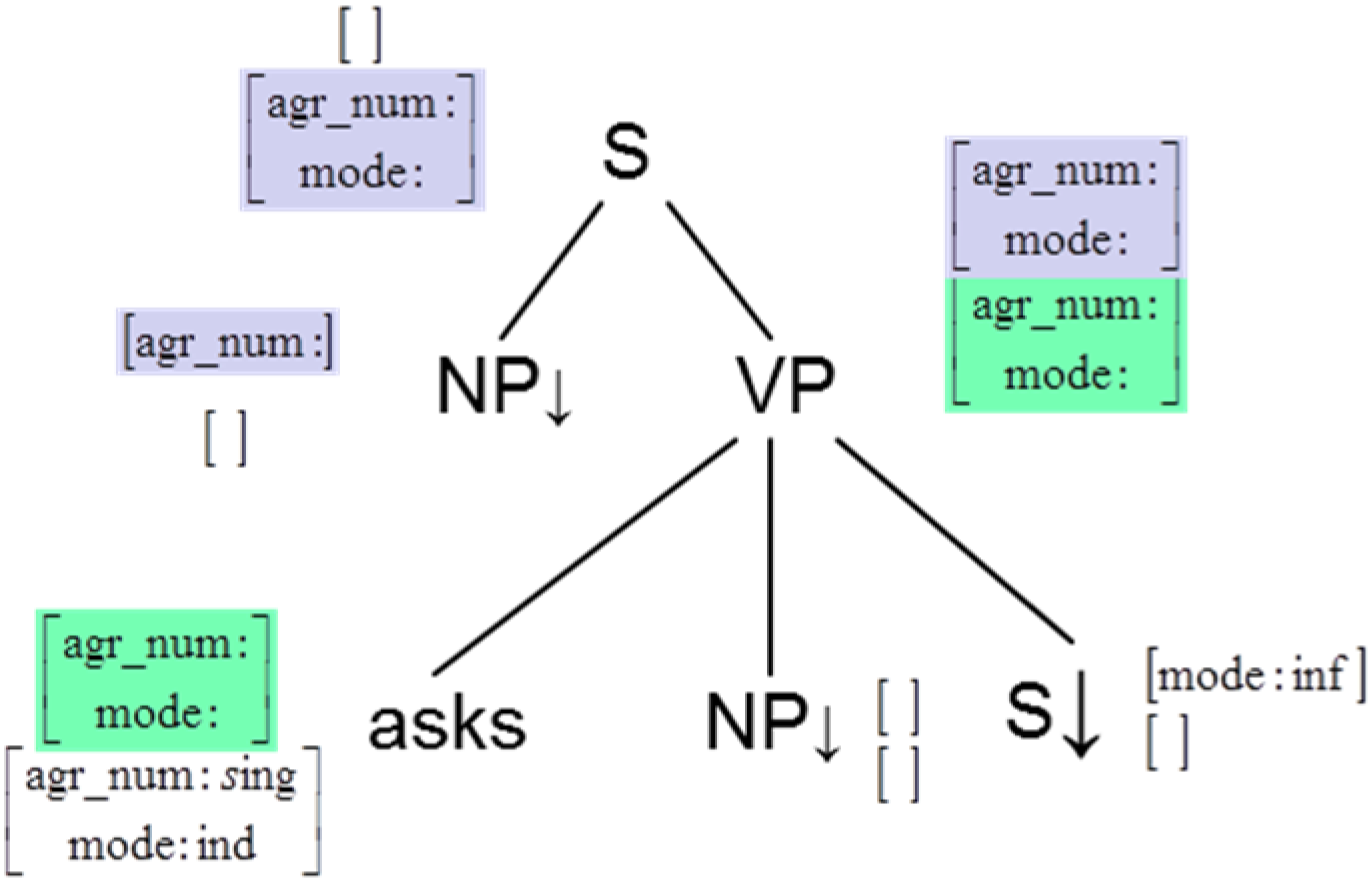
- A lexicalized grammar for target language
  - – Natural and intuitive to define lexical item's syntactic usage
  - – Easy to maintain and easy to extend
- The ability to simultaneously detect multiple ungrammatical types and their corresponding words based on the grammar

This paper

XTAG English Grammar

- The Feature-Based Lexicalized Tree Adjoining Grammars (FB-LTAG) formalism, released by UPENN in 2001
  - Each lexical item is associated with an elementary tree with attribute value matrixes (AVMs) to define its syntactic usage

Example for introducing XTAG



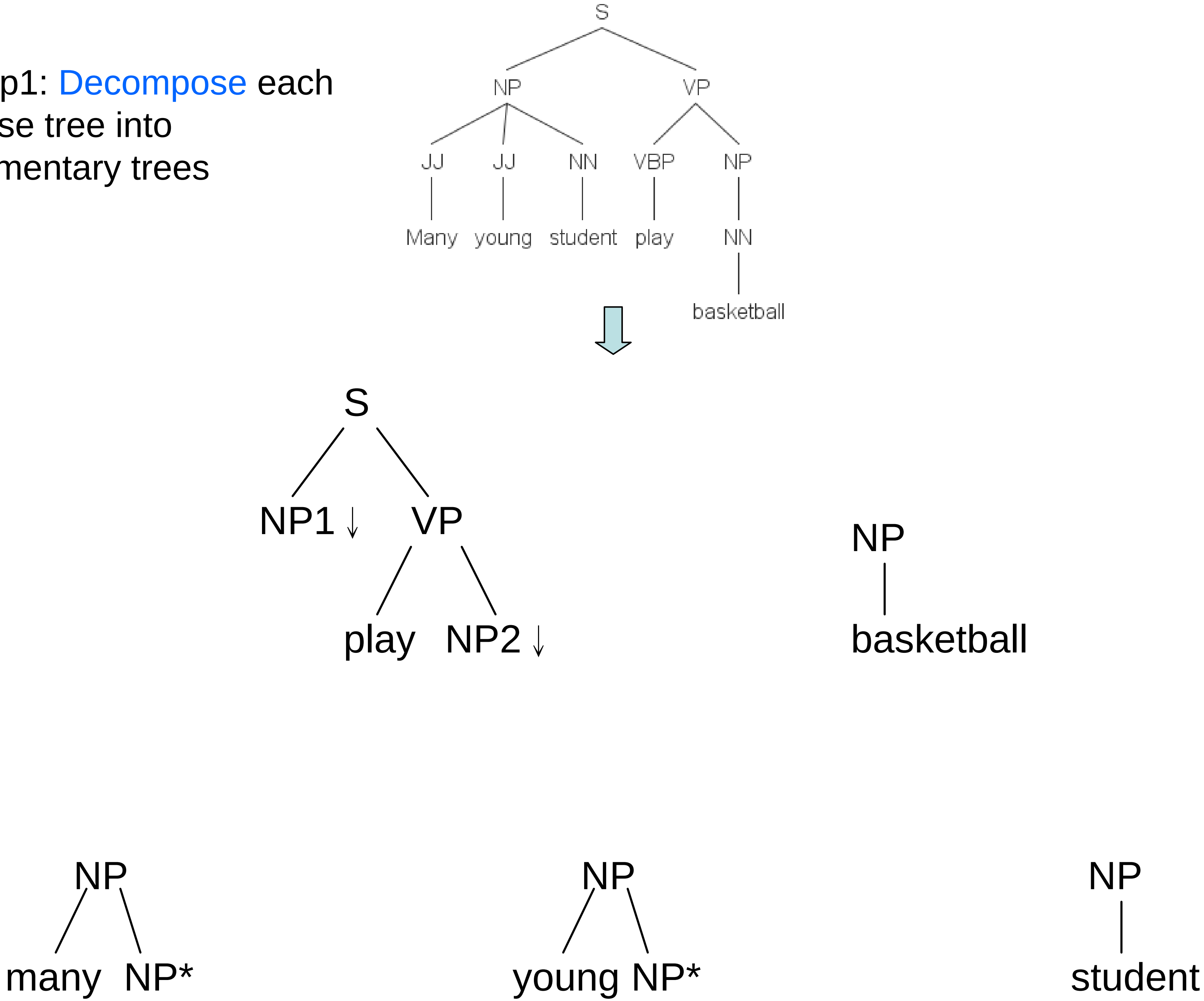
Elementary tree of “asks”
in FB-LTAG

Example:
He asks the girl to play basketball.

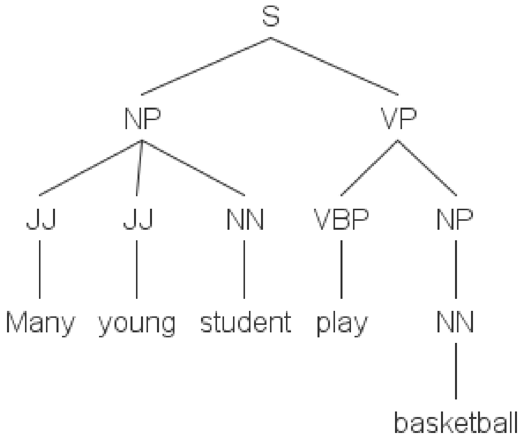
Procedure to detect ungrammatical types and words based on XTAG

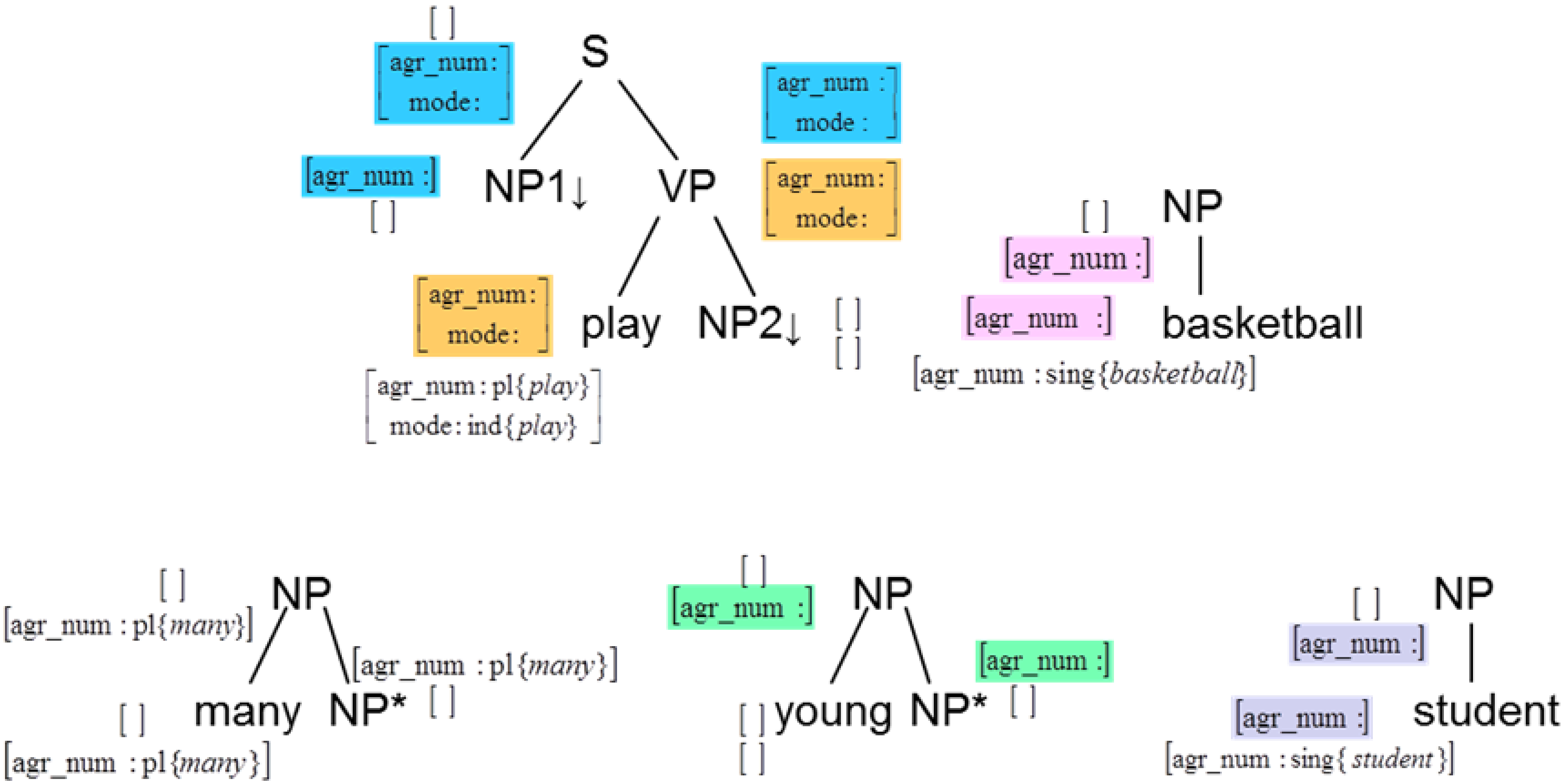
- Step1: Decompose each sentence hypothesis parse tree into elementary trees
- Step2: Associate each elementary tree with AVMs
- Step3: Reconstruct the original parse tree out of the elementary trees to check if AVMs contradict
  - Substitution and adjunction operations along with AVM unifications.
    - To simultaneously detect multiple error types and words, a new unification method is proposed

Step1: Decompose each parse tree into elementary trees

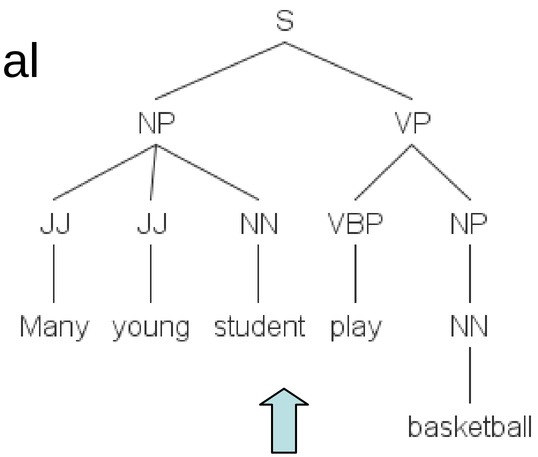


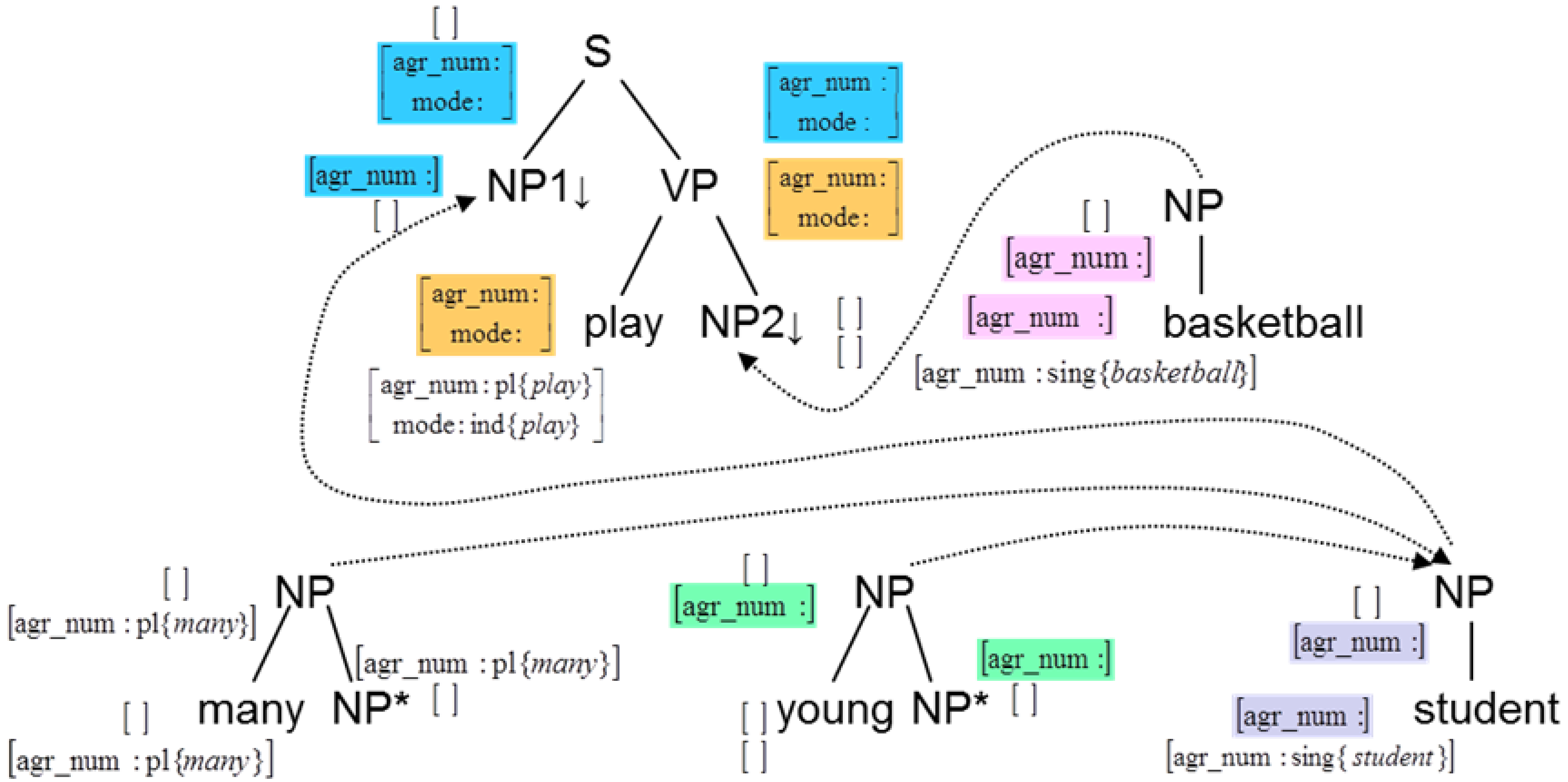
Step2: Associate each elementary tree with AVMs



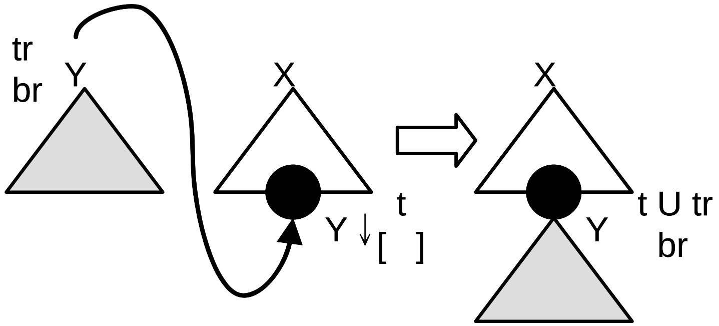


Step3: Reconstruct the original parse tree out of the elementary trees to check if AVMs contradict

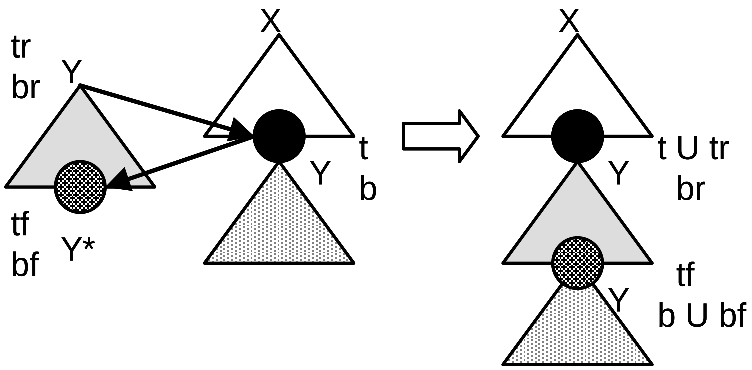




Substitution and adjunction operations along with AVM unifications.

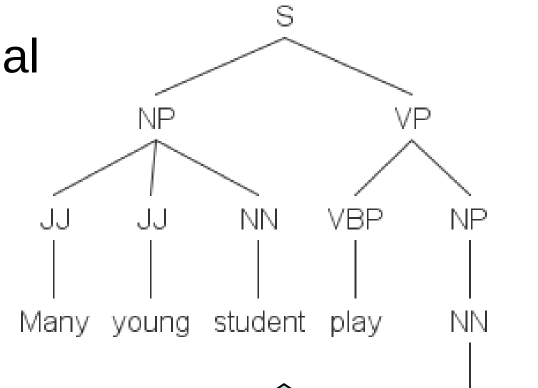


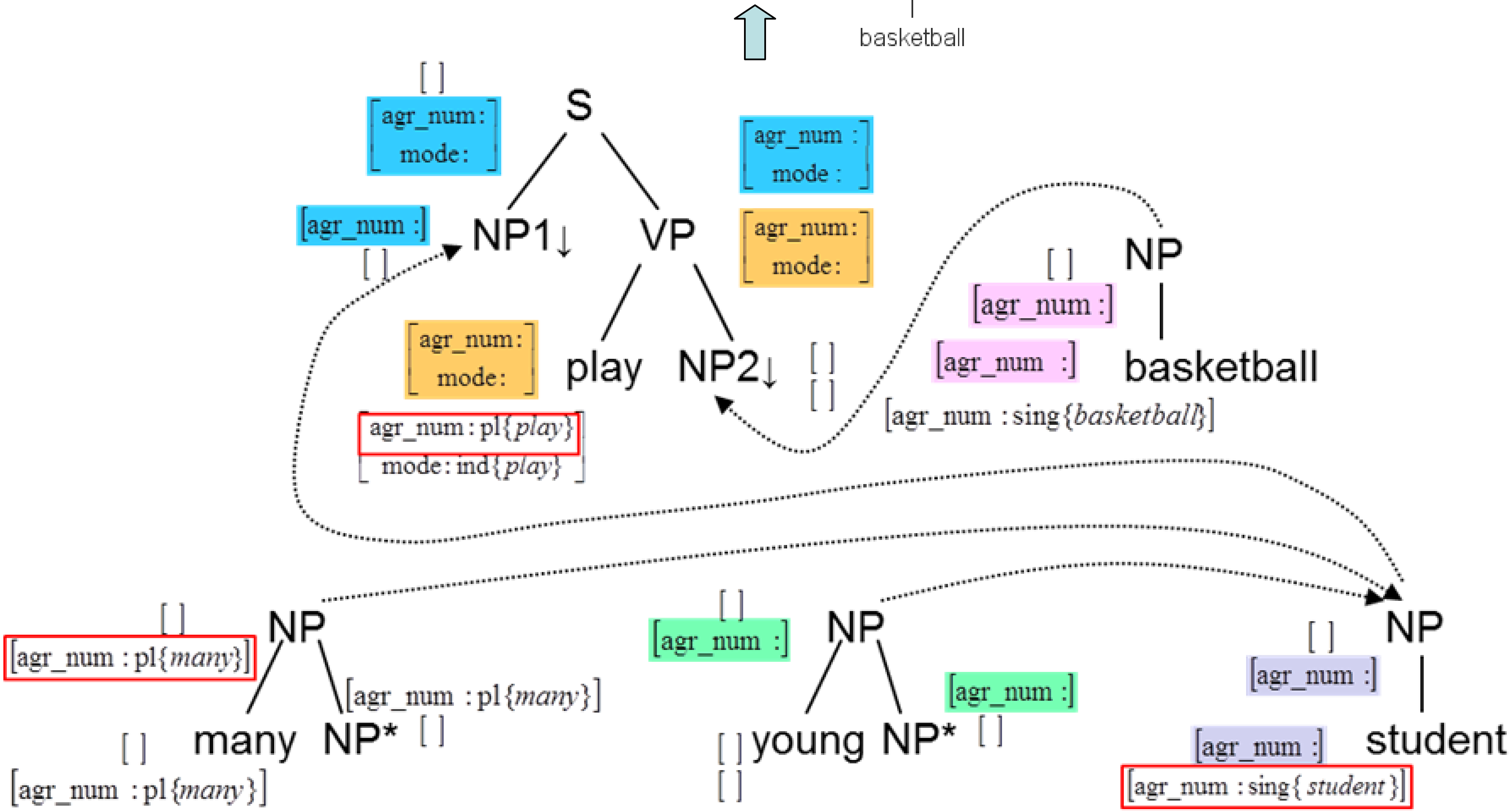
Substitution of FB-LTAG



Adjunction of FB-LTAG

Step3: Reconstruct the original parse tree out of the elementary trees to check if AVMs contradict





Traditional Unification Operations

- [f=x] unify [f=x] → [f=x]
- [f=x] unify [f=null] → [f=x]
- [f=null] unify [f=null] → [f=null]
- [f=x] unify [f=y] →fail

Example: “Many young student play basketball.”

[agr_num=plural]{many} U [agr_num=sing]{student} => fail {many,student}

How about “play”?

A new unification method:
fail propagation unification

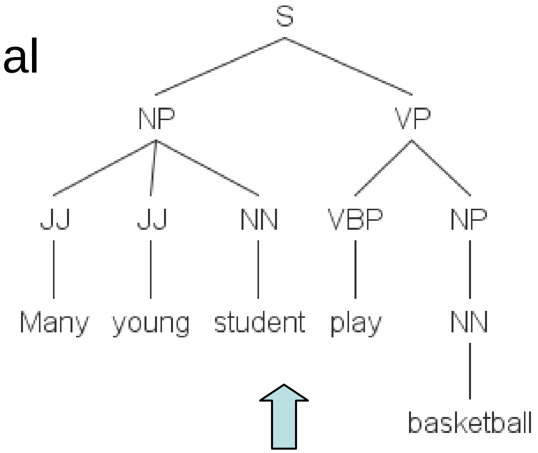
- [f=x] unify [f=x] → [f=x]
- [f=x] unify [f=null] → [f=x]
- [f=null] unify [f=null] → [f=null]
- [f=x] unify [f=y] → [f=fail]
- [f=fail] unify [f=null] →[f=fail]
- [f=fail] unify [f=y] →[f=fail]
- [f=fail] unify [f=fail] →[f=fail]

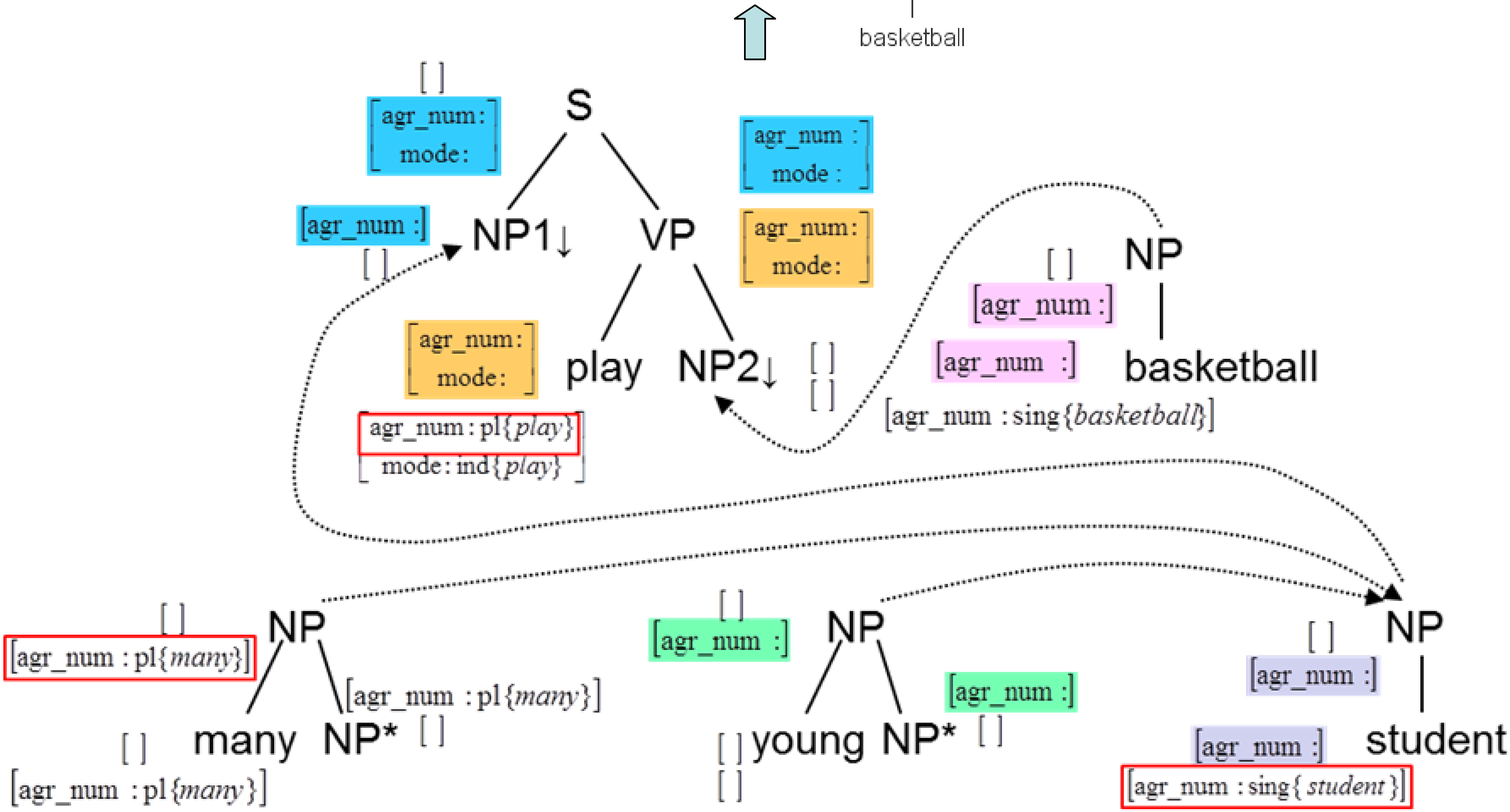
Example:

[agr_num=plural]{many} U [agr_num=sing]{student} => [agr_num =fail]{many,student}

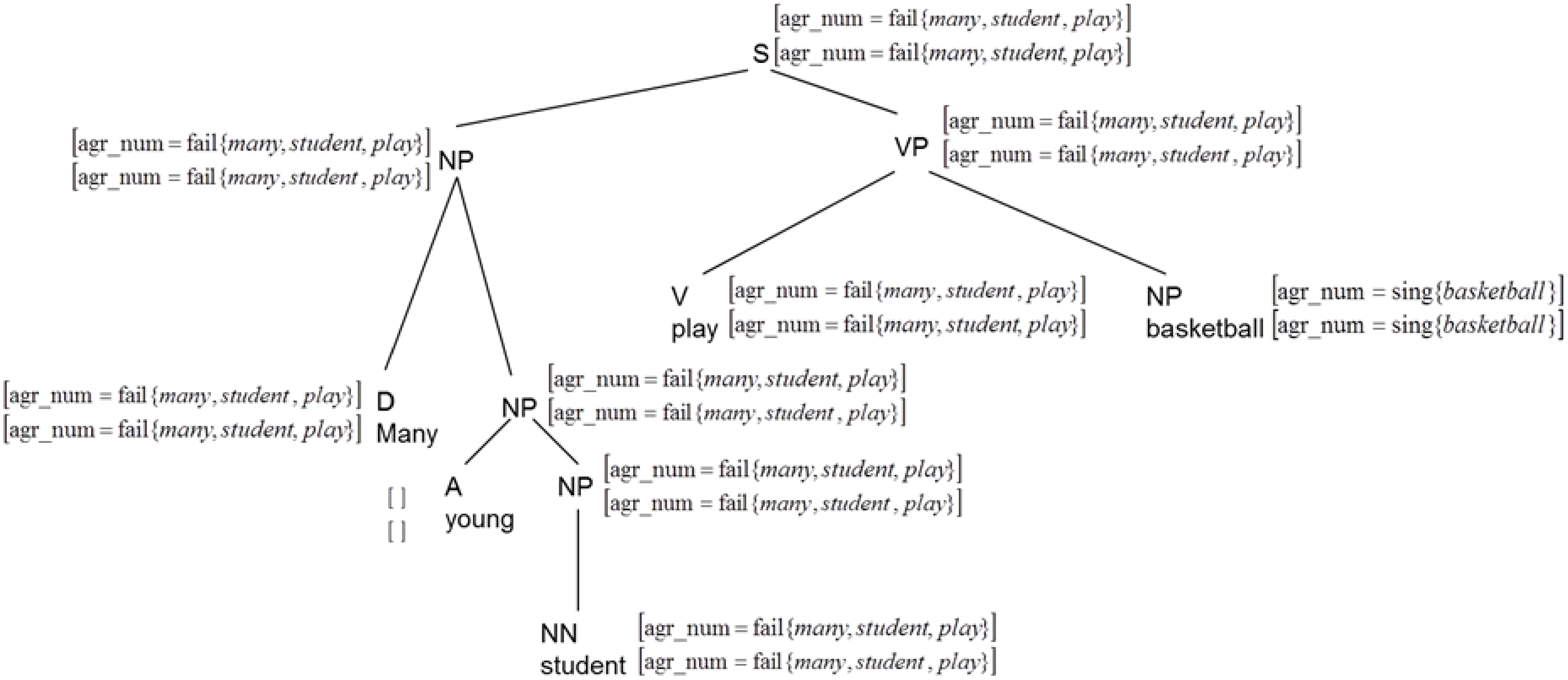
[agr_num=fail]{many, student} U [agr_num=plural]{play} => [agr_num =fail]{many,student,play}

Step3: Reconstruct the original parse tree out of the elementary trees to check if AVMs contradict

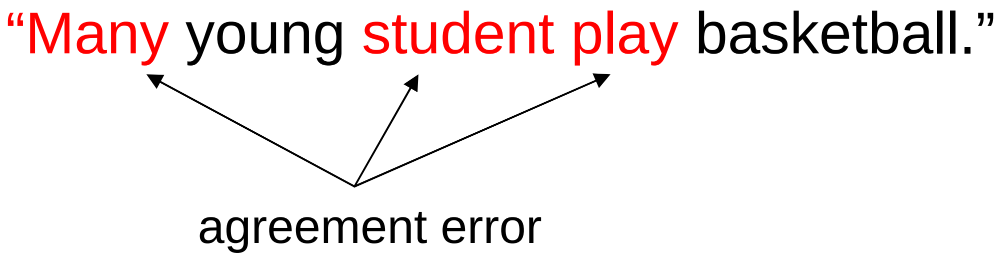




Example

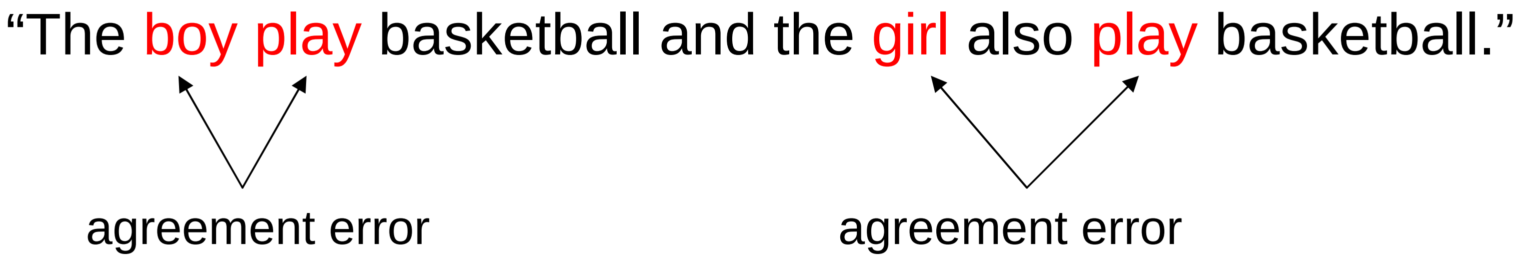

[agr_num = fail{many, student, play}]
S [agr_num = fail{many, student, play}]
[agr_num = fail{many, student, play}]
[agr_num = fail{many, student, play}] NP
[agr_num = fail{many, student, play}]
VP [agr_num = fail{many, student, play}]
V play [agr_num = fail{many, student, play}]
[agr_num = fail{many, student, play}]
NP basketball [agr_num = sing{basketball}]
[agr_num = sing{basketball}]
[agr_num = fail{many, student, play}] D Many
[agr_num = fail{many, student, play}]
NP [agr_num = fail{many, student, play}]
[agr_num = fail{many, student, play}]
[ ] A young
[ ]
NP [agr_num = fail{many, student, play}]
[agr_num = fail{many, student, play}]
NN student [agr_num = fail{many, student, play}]
[agr_num = fail{many, student, play}]

Our Goal of Detection



"Many young student play basketball."

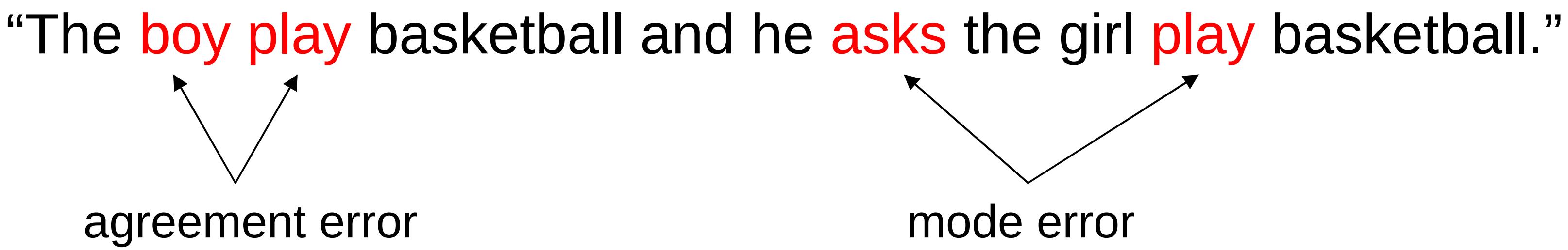
agreement error



"The boy play basketball and the girl also play basketball."

agreement error

agreement error



"The boy play basketball and he asks the girl play basketball."

agreement error

mode error

Syntactic Error Correction

- To date, we have developed two simple mechanism to handle part of the detected situations
  - Feature value voting

    Ex: “Many young student play basketball.”

    ↓

    students
  - For equal votes, we tend to correct nouns

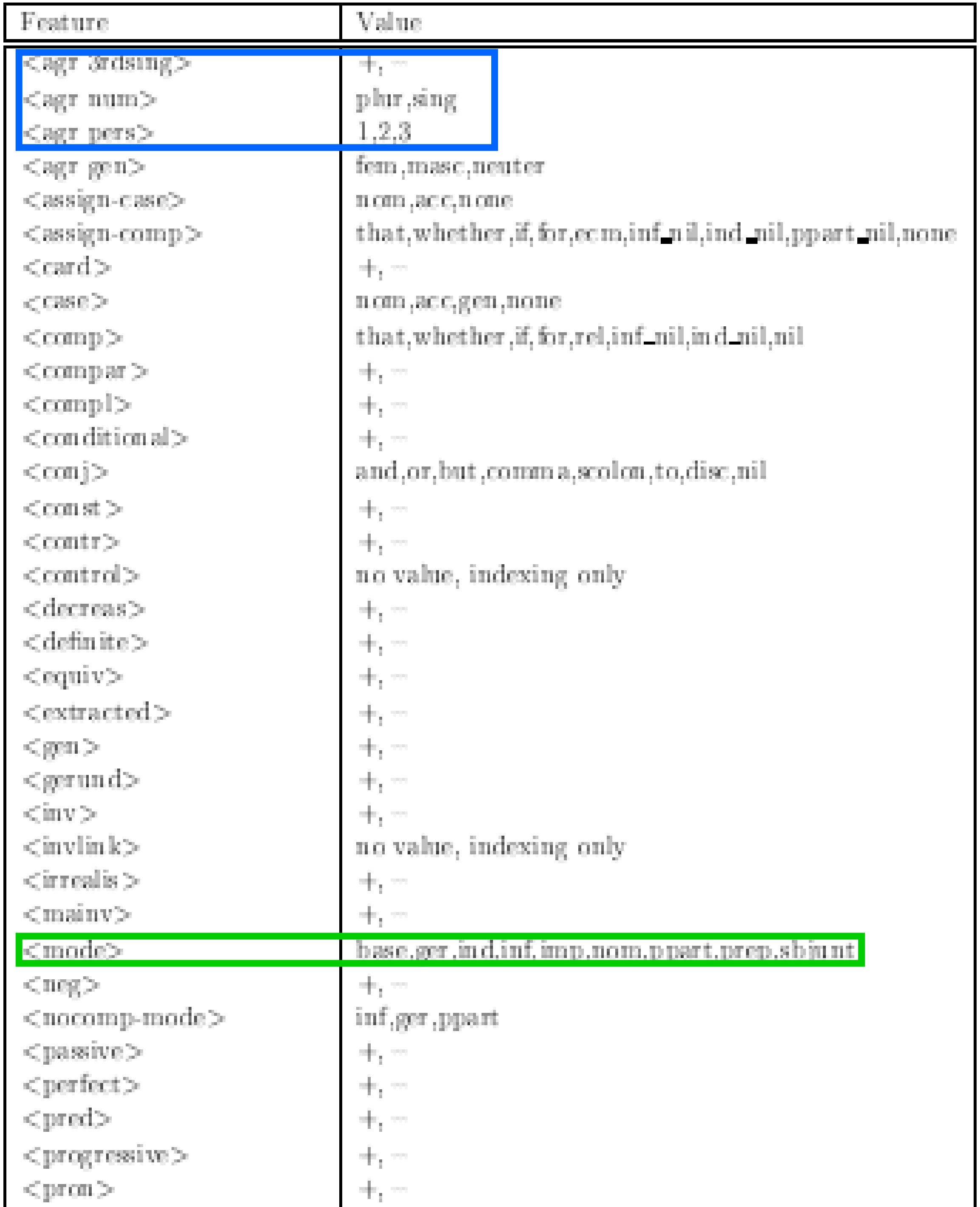
Current Implementation:

For Error Detection:

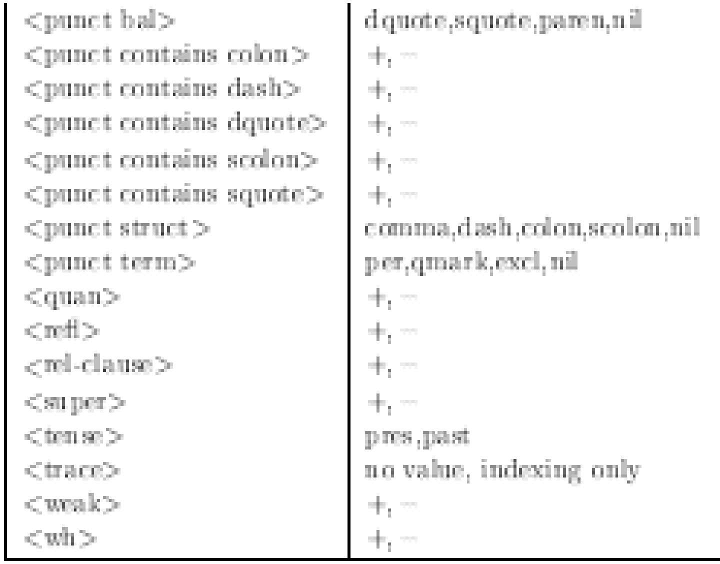
- agreement features
- mode feature

For Error Correction:

- agreement features



Feature	Value
<agr 3rdsing>	+, −
<agr num>	plur,sing
<agr pers>	1,2,3
<agr gen>	fem,masc,neuter
<assign-case>	nom,acc,none
<assign-comp>	that,whether,if,for,ecm,inf_nil,ind_nil,ppart_nil,none
<card>	+, −
<case>	nom,acc,gen,none
<comp>	that,whether,if,for,rel,inf_nil,ind_nil,nil
<compar>	+, −
<compl>	+, −
<conditional>	+, −
<conj>	and,or,but,comma,scolon,to,disc,nil
<const>	+, −
<contr>	+, −
<control>	no value, indexing only
<decreas>	+, −
<definite>	+, −
<equiv>	+, −
<extracted>	+, −
<gen>	+, −
<gerund>	+, −
<inv>	+, −
<invlink>	no value, indexing only
<irrealis>	+, −
<mainv>	+, −
<mode>	base,ger,ind,inf,imp,nom,ppart,prep,sbjunct
<neg>	+, −
<nocomp-mode>	inf,ger,ppart
<passive>	+, −
<perfect>	+, −
<pred>	+, −
<progressive>	+, −
<pron>	+, −



<punct bal>	dquote,squote,paren,nil
<punct contains colon>	+, −
<punct contains dash>	+, −
<punct contains dquote>	+, −
<punct contains scolon>	+, −
<punct contains squote>	+, −
<punct struct>	comma,dash,colon,scolon,nil
<punct term>	per,qmark,excl,nil
<quan>	+, −
<refl>	+, −
<rel-clause>	+, −
<super>	+, −
<tense>	pres,past
<trace>	no value, indexing only
<weak>	+, −
<wh>	+, −

Experiment

- Setting
  - 422 translation sentences of six Chinese-English MT systems from the DARPA Global Autonomous Language Exploitation (GALE) 2008 evaluation

MT System name	Approach
NRC	phrase-based SMT
RWTH-PBT	phrase-based SMT
RWTH-PBT-AML	phrase-based SMT + source reordering
RWTH-PBT-JX	phrase-based SMT + Chinese word segmentation
RWTH-PBT-SH	phrase-based SMT + source reordering + rescoring
SRI-HPBT	hierarchical phrase-based SMT

Result

MT system name	Detected sentences	Corrected sentences	Bleu for all 422 sentences (before correct)	Bleu for all 422 sentences (after correct)	Bleu for only corrected sentences (before correct)	Bleu for only corrected sentences (after correct)
NRC	23	9	32.99	32.99	26.75	27.80
RWTH-PBT	23	14	27.95	27.97	22.08	23.03
RWTH-PBT-AML	18	7	34.40	34.41	32.13	32.67
RWTH-PBT-JX	25	14	32.96	32.99	31.49	32.17
RWTH-PBT-SH	30	11	34.64	34.68	29.31	30.61
SRI-HPBT	18	8	34.13	34.14	29.15	28.83

agreement features
mode feature

agreement features

Result

MT system name	Detected sentences	Corrected sentences	Bleu for all 422 sentences (before correct)	Bleu for all 422 sentences (after correct)	Bleu for only corrected sentences (before correct)	Bleu for only corrected sentences (after correct)
NRC	23	9	32.99	32.99	26.75	27.80
RWTH-PBT	23	14	27.95	27.97	22.08	23.03
RWTH-PBT-AML	18	7	34.40	34.41	32.13	32.67
RWTH-PBT-JX	25	14	32.96	32.99	31.49	32.17
RWTH-PBT-SH	30	11	34.64	34.68	29.31	30.61
SRI-HPBT	18	8	34.13	34.14	29.15	28.83

Result

MT system name	Detected sentences	Corrected sentences	Bleu for all 422 sentences (before correct)	Bleu for all 422 sentences (after correct)	Bleu for only corrected sentences (before correct)	Bleu for only corrected sentences (after correct)
NRC	23	9	32.99	32.99	26.75	27.80
RWTH-PBT	23	14	27.95	27.97	22.08	23.03
RWTH-PBT-AML	18	7	34.40	34.41	32.13	32.67
RWTH-PBT-JX	25	14	32.96	32.99	31.49	32.17
RWTH-PBT-SH	30	11	34.64	34.68	29.31	30.61
SRI-HPBT	18	8	34.13	34.14	29.15	28.83

Conclusion

- We present a novel post-editing approach for MT which features:
  - The use of a lexicalized grammar for target language
  - A framework to simultaneously detect multiple ungrammatical types and their corresponding words
    - A new unification method - fail propagation unification is proposed
  - The Initial attempt to correct errors based on the detected information.

Q/A