A rule-based framework for sequence labeling tasks and its application to Vietnamese NLP

Dat Quoc Nguyen

Introduction

- Sequence labeling tasks:
  - Part-Of-Speech (POS) tagging
    - Assign a tag representing its lexical category to each word: ``/`` *We/PRP do/VBP n't/RB have/VB passive/JJ readers/NNS ./. ”/”*
  - Named entity recognition, chunking
- ML-based approaches:
    - Maximum Entropy, SVM, Perceptron learning, CRF,…
      - Involve a quite complex configuration over feature extraction process
    - Neural network-based models:
      - Hyper-parameter tuning, architecture engineering

Introduction

- ML-based approaches:
  - Could be time-consuming in the learning process
  - Require a powerful computer to train labeling models
  - The training time in minutes reported in (Mueller et al., 2013) for POS+MORPH tagging on a machine of two Hexa-Core Intel Xeon X5680 CPUs with 3,33 GHz and 6 cores each and 144 GB of memory. SVMT: SVMTool, Morf: Morfette, CRFS: CRFSuite

Language	#sent	#tags	SVMT	Morf	CRFS
German	40,474	681	1,649	286	1,295
Czech	38,727	1,811	2,454	539	9,274
Spanish	14,329	303	64	63	69

Mueller, T., Schmid, H., Sch¨utze, H., 2013. Efficient Higher-Order CRFs for Morphological Tagging. In: Proceedings of EMNLP'13. pp. 322–332.

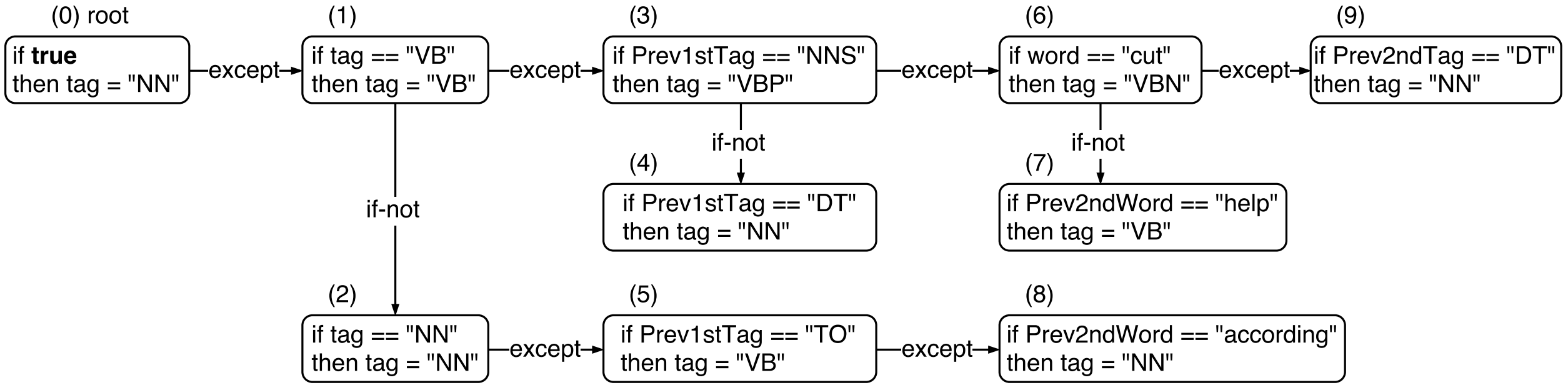
Introduction

- The well-known transformation rule-based tagger of Eric Brill:
  - Difficulty to control the interactions among a large numbers of rules: a rule will change the outputs of all the preceding rules
- Our approach:
  - Using Single Classification Ripple Down Rules (SCRDR) tree (Compton and Jansen, 1990) to control the interactions between rules
  - Fast in terms of training time and labeling process
  - Competitive results to other ML-based models

Brill, E.: Transformation-based error-driven learning and natural language processing: a case study in part-of-speech tagging. Computational Linguistics **21(4)** (1995) 543–565

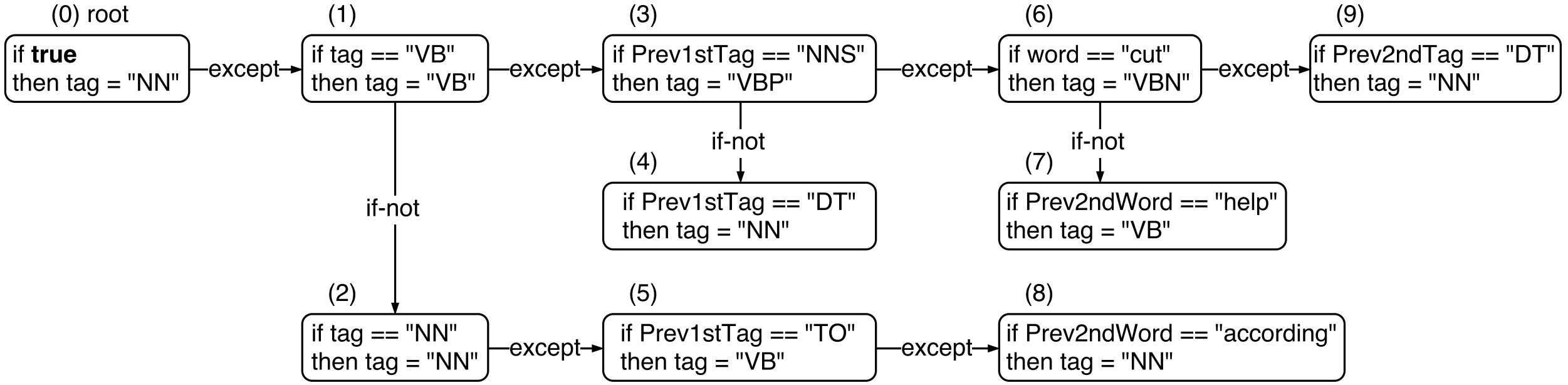
Single Classification Ripple Down Rules

- A SCRDR tree (Compton and Jansen, 1990):
  - a binary tree with only two unique types of edges “except” and “if-not”
  - every node is associated with a rule in a form of “**if** *condition* **then** *conclusion*”



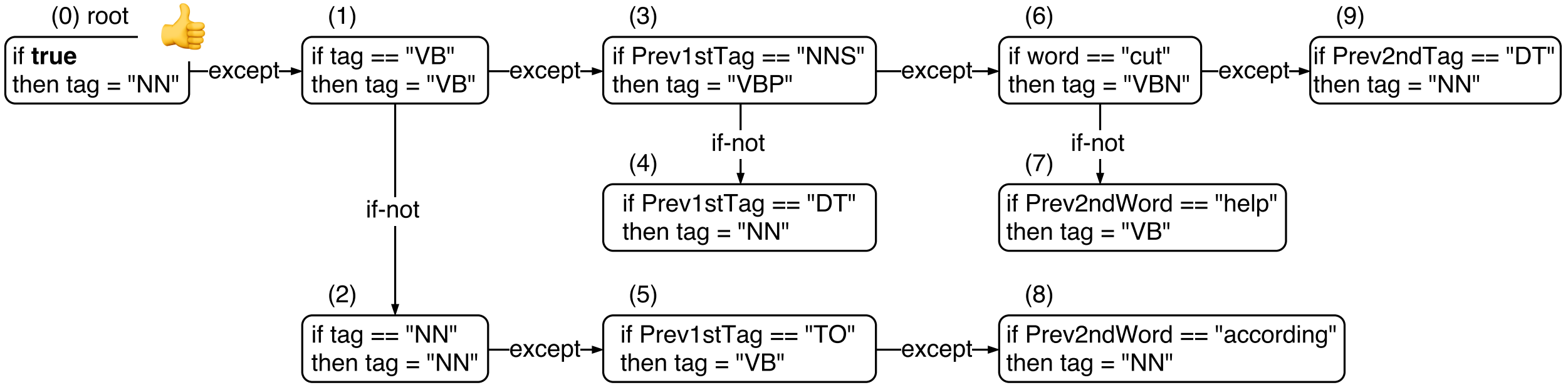
Single Classification Ripple Down Rules

- Each case to be evaluated starts at the root node and ripples down as follows:
  - If the case satisfies the condition of a current node's rule, the case is then passed on to the current node's "except" child if this "except" child exists
  - Otherwise, if the case is then passed on to the current node's "if-not" child
  - The conclusion returned by the tree is the conclusion of the last satisfied rule in the evaluation path to a leaf node



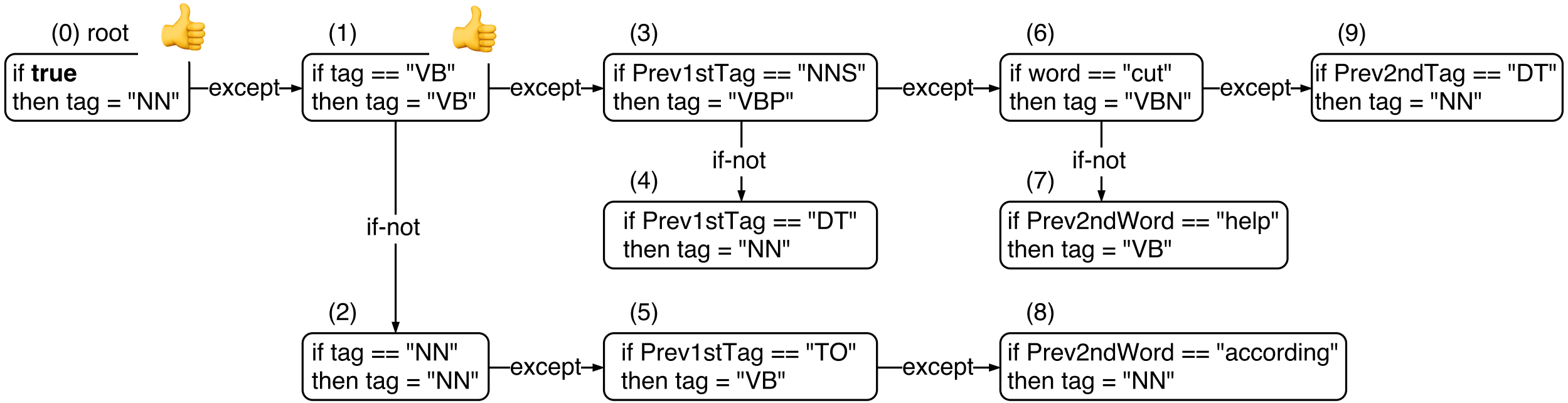
Single Classification Ripple Down Rules

as/IN investors/NNS anticipate/VB a/DT recovery/NN



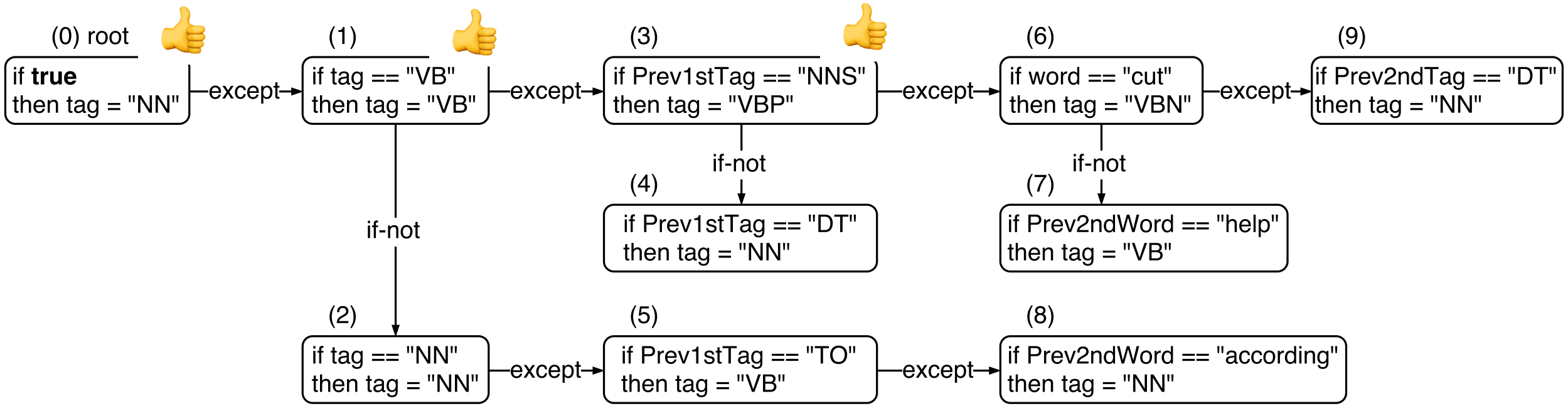
Single Classification Ripple Down Rules

as/IN investors/NNS anticipate/**VB** a/DT recovery/NN



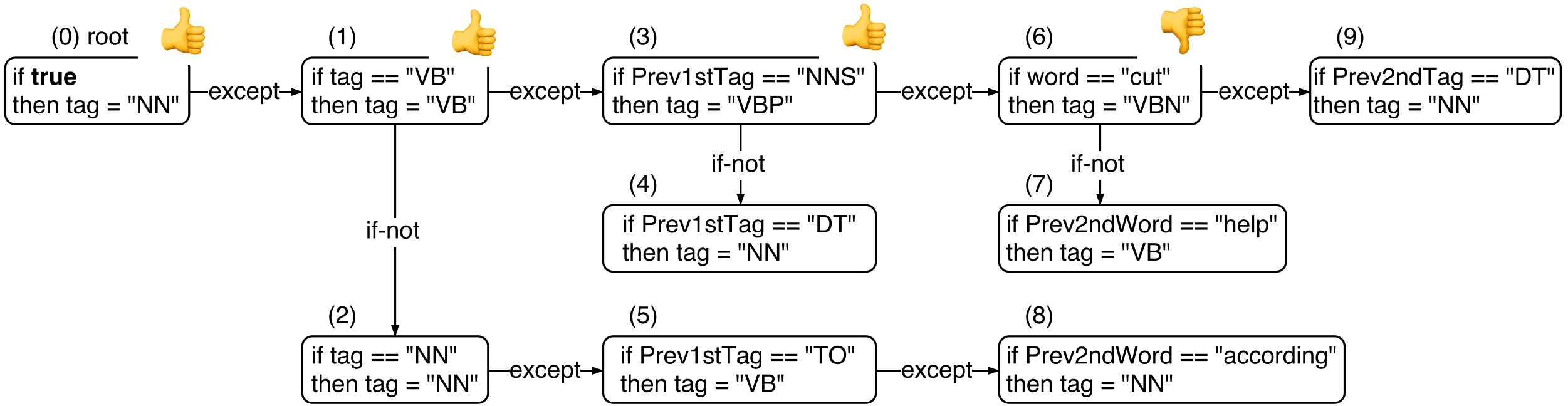
Single Classification Ripple Down Rules

as/IN investors/**NNS** anticipate/**VB** a/DT recovery/NN

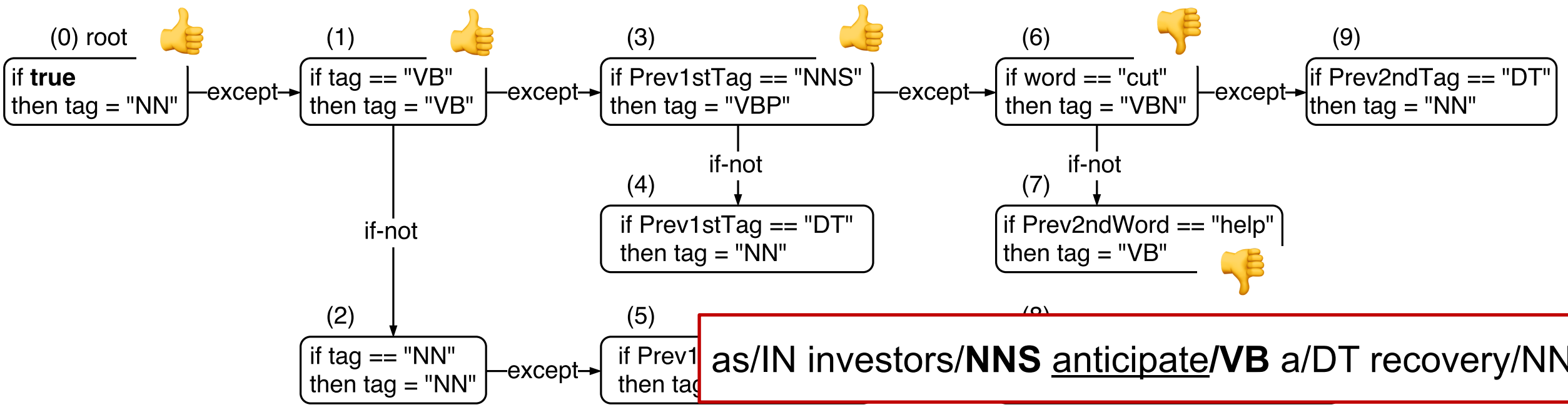


Single Classification Ripple Down Rules

as/IN investors/**NNS** anticipate/**VB** a/DT recovery/NN

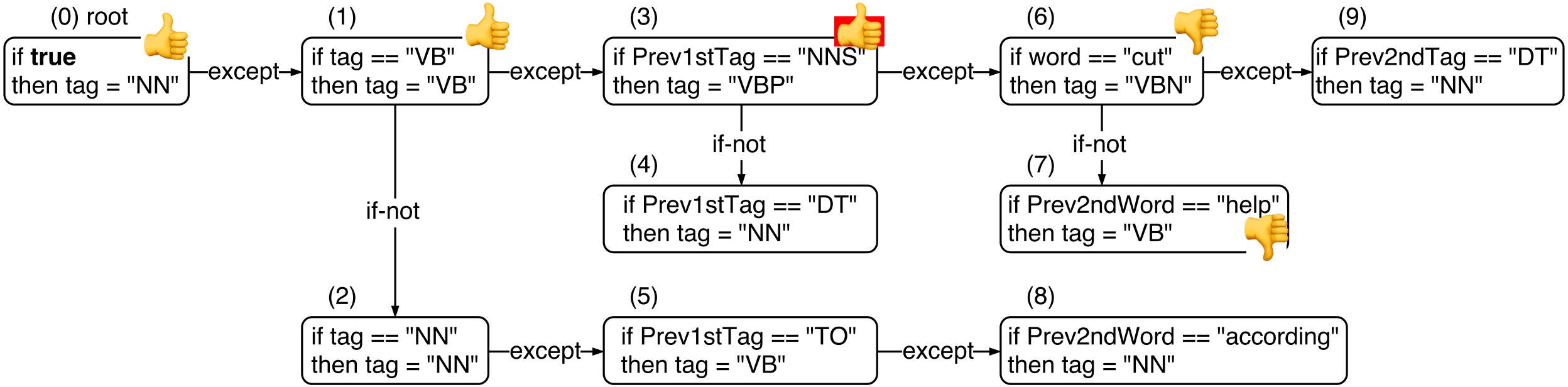


Single Classification Ripple Down Rules

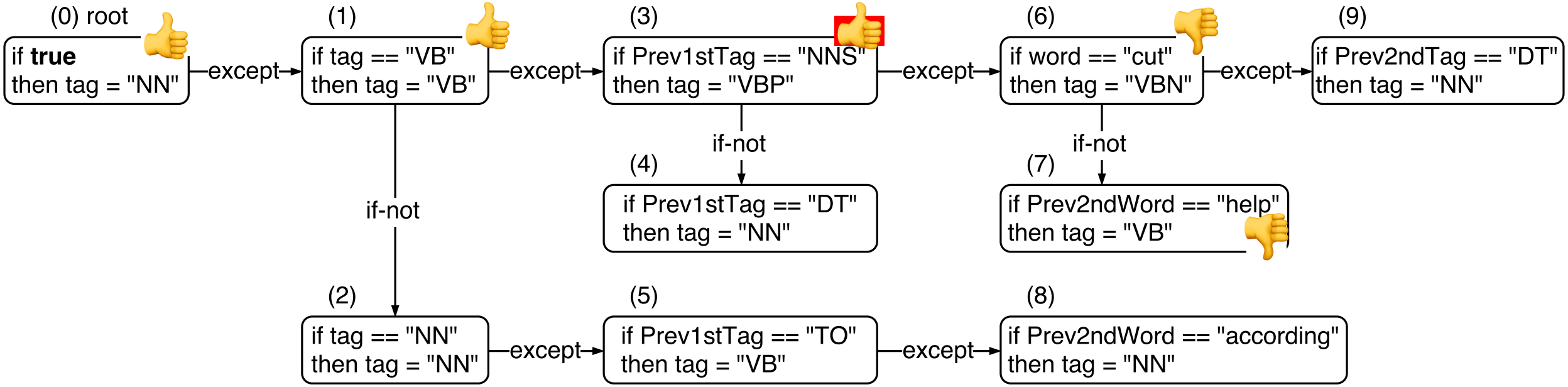


Single Classification Ripple Down Rules

- as/IN investors/**NNS** anticipate/**VB** a/DT recovery/NN
  - evaluation path (0)-(1)-**(3)**-(6)- (7) with the last satisfied node **(3)**
  - “VBP” should be the POS tag of the word “anticipate”



Single Classification Ripple Down Rules



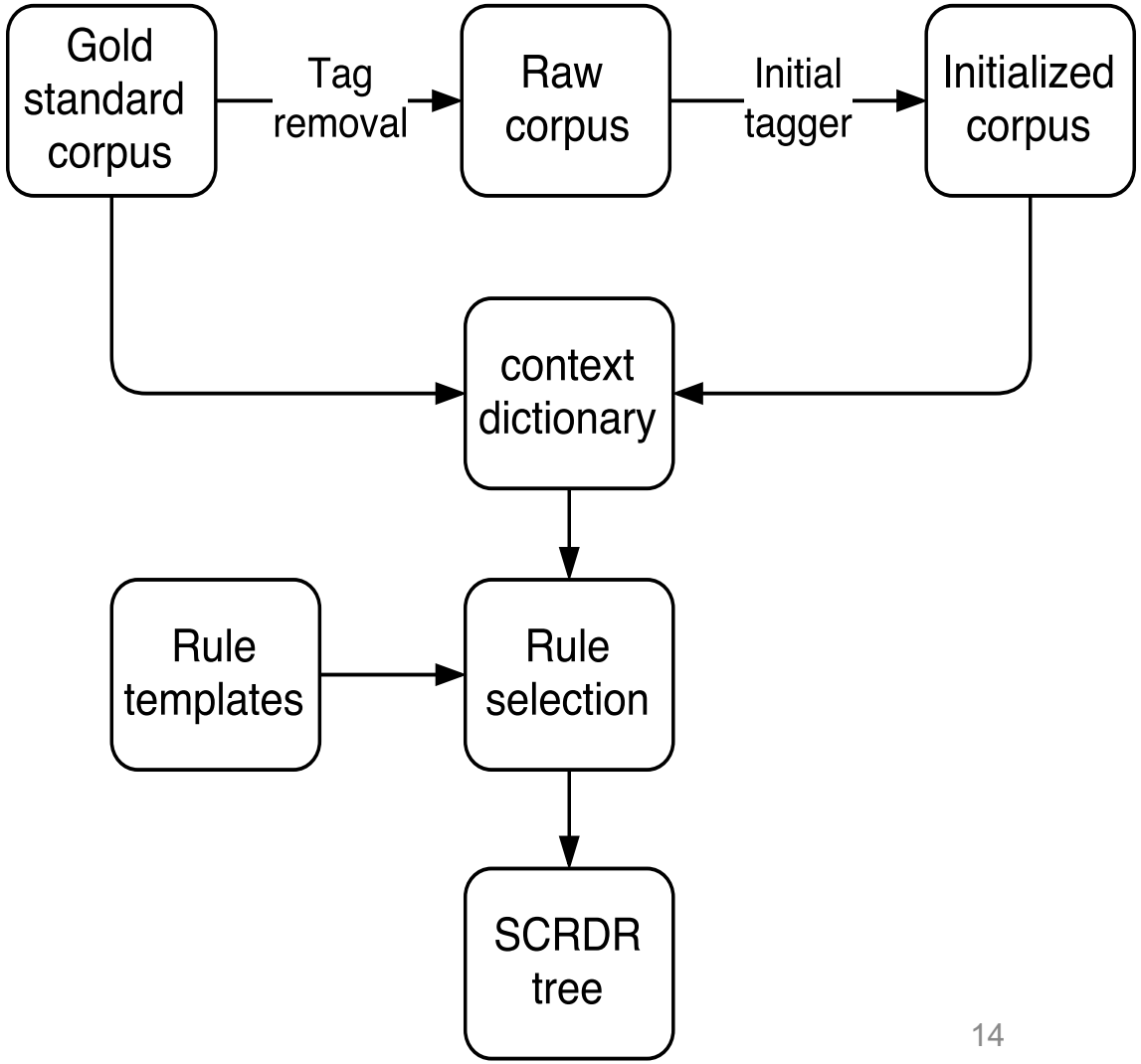
⇒To correct a wrong conclusion returned for a given case, a new node containing a new exception rule may be attached to the last node in the evaluation path

⇒If the last node is the fired node given the case, the new node is added as its child with the “except” edge

⇒Otherwise, the new node is attached with the “if-not” edge

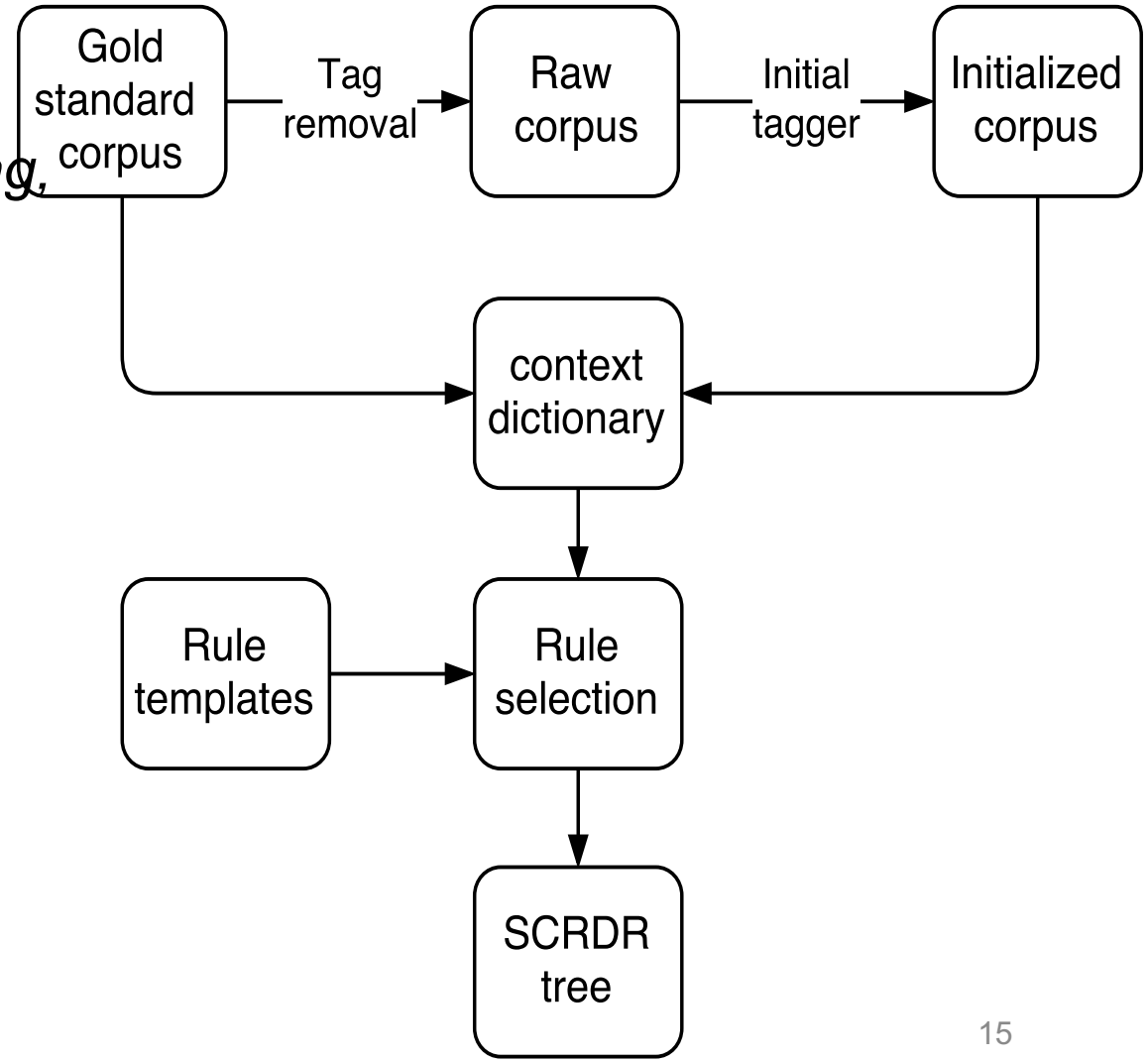
Rule-based framework for sequence labeling

- Gold standard corpus:
  as/IN investors/NNS anticipate/VBP a/DT recovery/NN
- Raw corpus:
  as investors anticipate a recovery
- Initialized corpus:
  as/IN investors/NNS anticipate/VB a/DT recovery/NN



Rule-based framework for sequence labeling

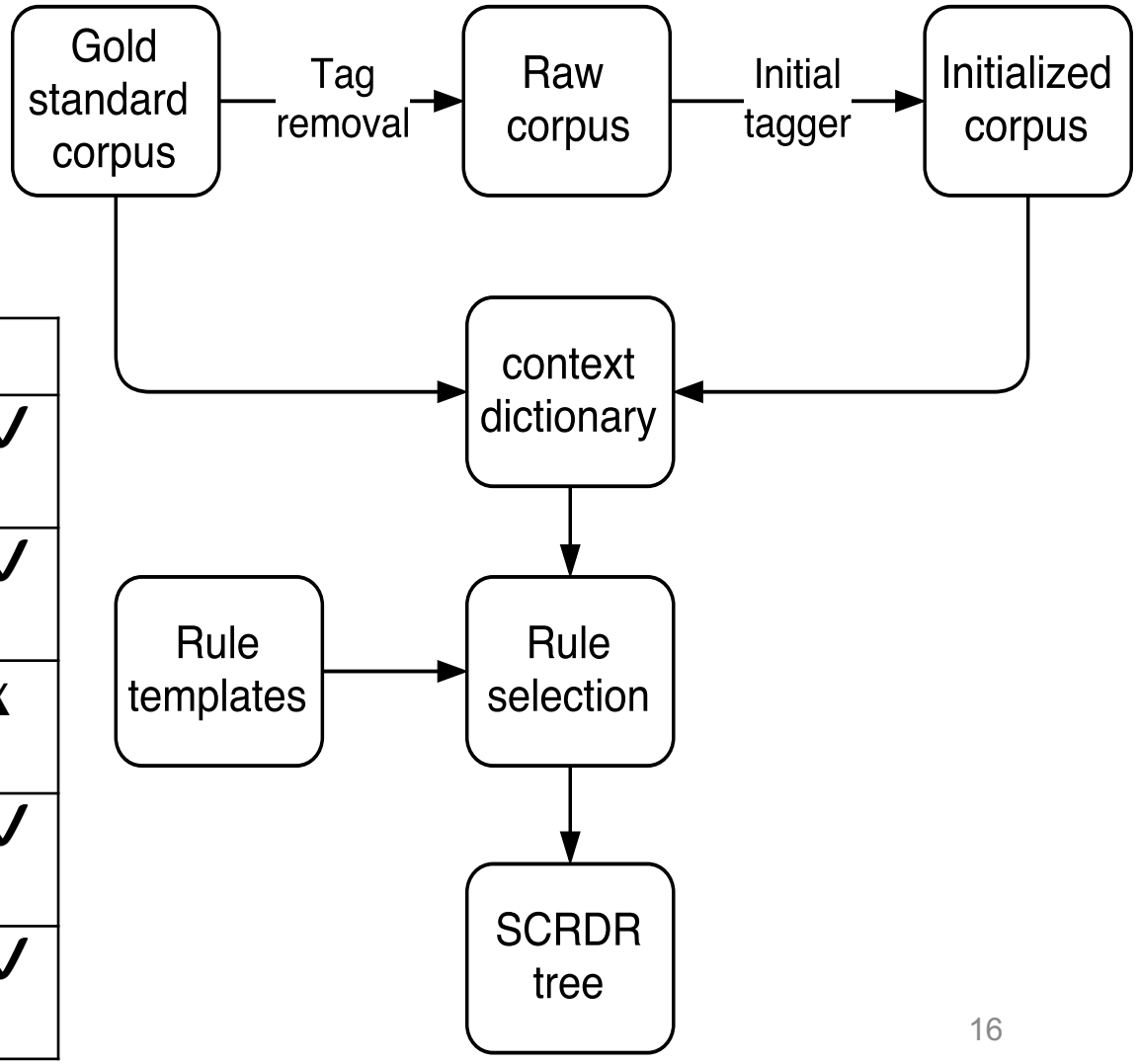
- Context dictionary (key, value):
- Tuple (*Previous-2nd-word, Previous-2nd-tag, Previous-1st-word, Previous-1st-tag, word, tag, Next- 1st-word, Next-1st-tag, Next-2nd-word, Next-2nd-tag, last-2-characters, last-3-characters, last-4-characters*) as dictionary key, extracted from the initialized corpus
- Gold label as value, extracted from the gold standard corpus



Rule-based framework for sequence labeling

- Gold standard corpus: as/IN investors/NNS anticipate/VBP a/DT recovery/NN
- Initialized corpus: as/IN investors/NNS anticipate/VB a/DT recovery/NN

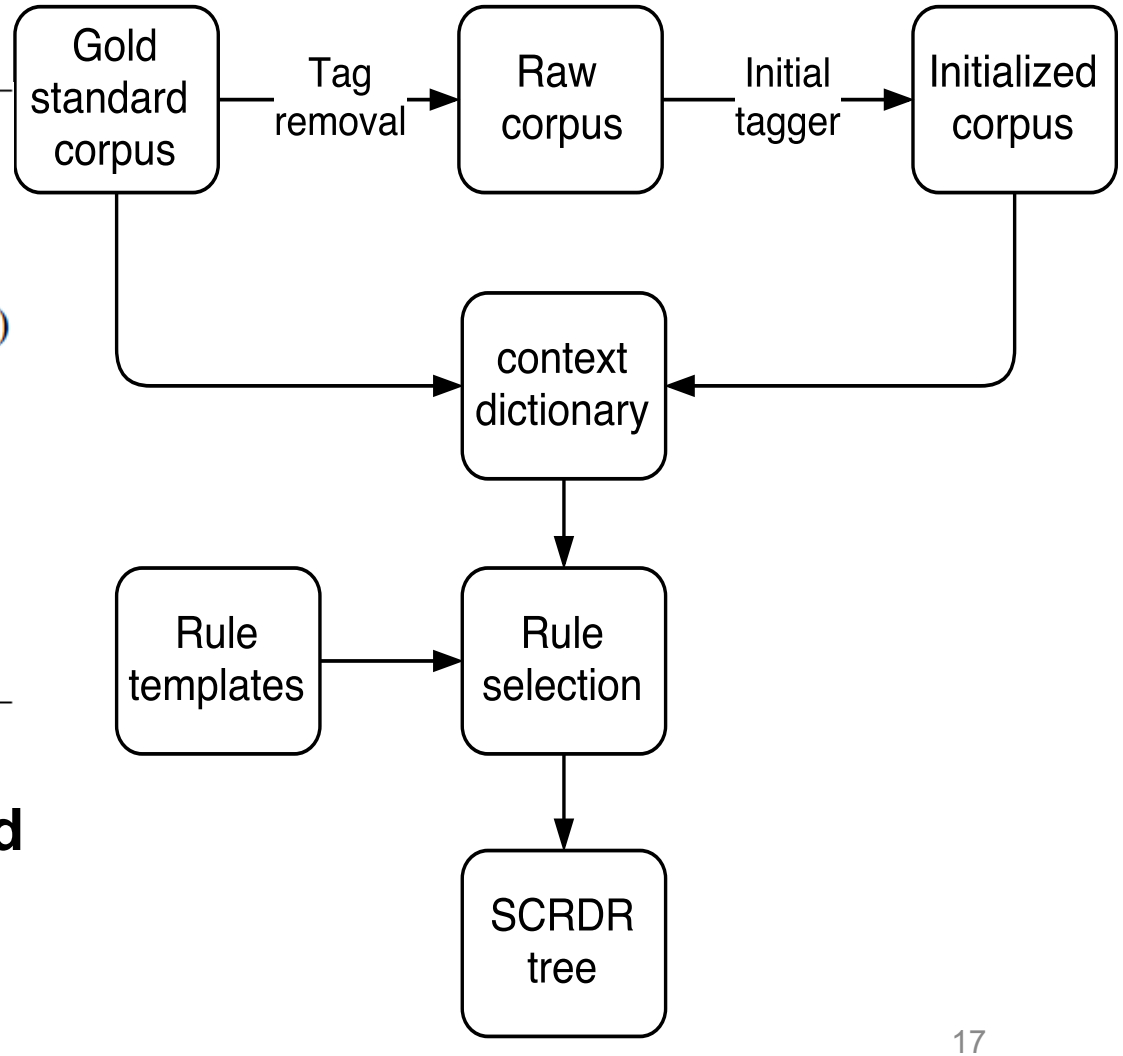
Tuple as key	**Value**	
(“”, “”, “”, “”, **as, IN**, investors, NNS, anticipate, VB, as, “”, “”)	IN	✓
(“”, “”, as, IN, **investors, NNS**, anticipate, VB, a, DT, “rs”, “ors”, “tors”)	NNS	✓
(as, IN, investors, NNS, **anticipate, VB**, a, DT, recovery, NN, te, ate, pate)	VBP	X
(investors, NNS, anticipate, VB, **a, DT**, recovery, NN, “”, “”, “”, “”, “”)	DT	✓
(anticipate, VB, a, DT, **recovery, NN**, “”, “”, “”, “”, “ry”, “ery”, “very”)	NN	✓



Rule-based framework for sequence labeling

- Short descriptions of our rule templates

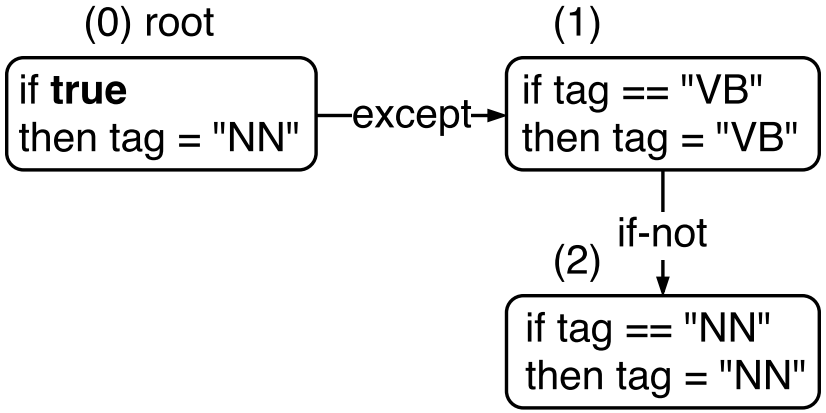
Words	$w_{-2}, w_{-1}, w_0, w_{+1}, w_{+2}$
Word bigrams	$(w_{-2}, w_0), (w_{-1}, w_0), (w_{-1}, w_{+1}), (w_0, w_{+1}), (w_0, w_{+2})$
Word trigrams	$(w_{-2}, w_{-1}, w_0), (w_{-1}, w_0, w_{+1}), (w_0, w_{+1}, w_{+2})$
tags	$p_{-2}, p_{-1}, p_0, p_{+1}, p_{+2}$
tag bigrams	$(p_{-2}, p_{-1}), (p_{-1}, p_{+1}), (p_{+1}, p_{+2})$
Combined	$(p_{-1}, w_0), (w_0, p_{+1}), (p_{-1}, w_0, p_{+1}), (p_{-2}, p_{-1}, w_0), (w_0, p_{+1}, p_{+2})$
Suffixes	$c_{n-1}c_n, c_{n-2}c_{n-1}c_n, c_{n-3}c_{n-2}c_{n-1}c_n$



- (w_{-1}, w_{+1}) represents the rule template "IF Previous-1st-word == tuple.**Previous-1st-word** && Next-1st-word == tuple.**Next-1st-word** THEN tag = **gold-tag**"

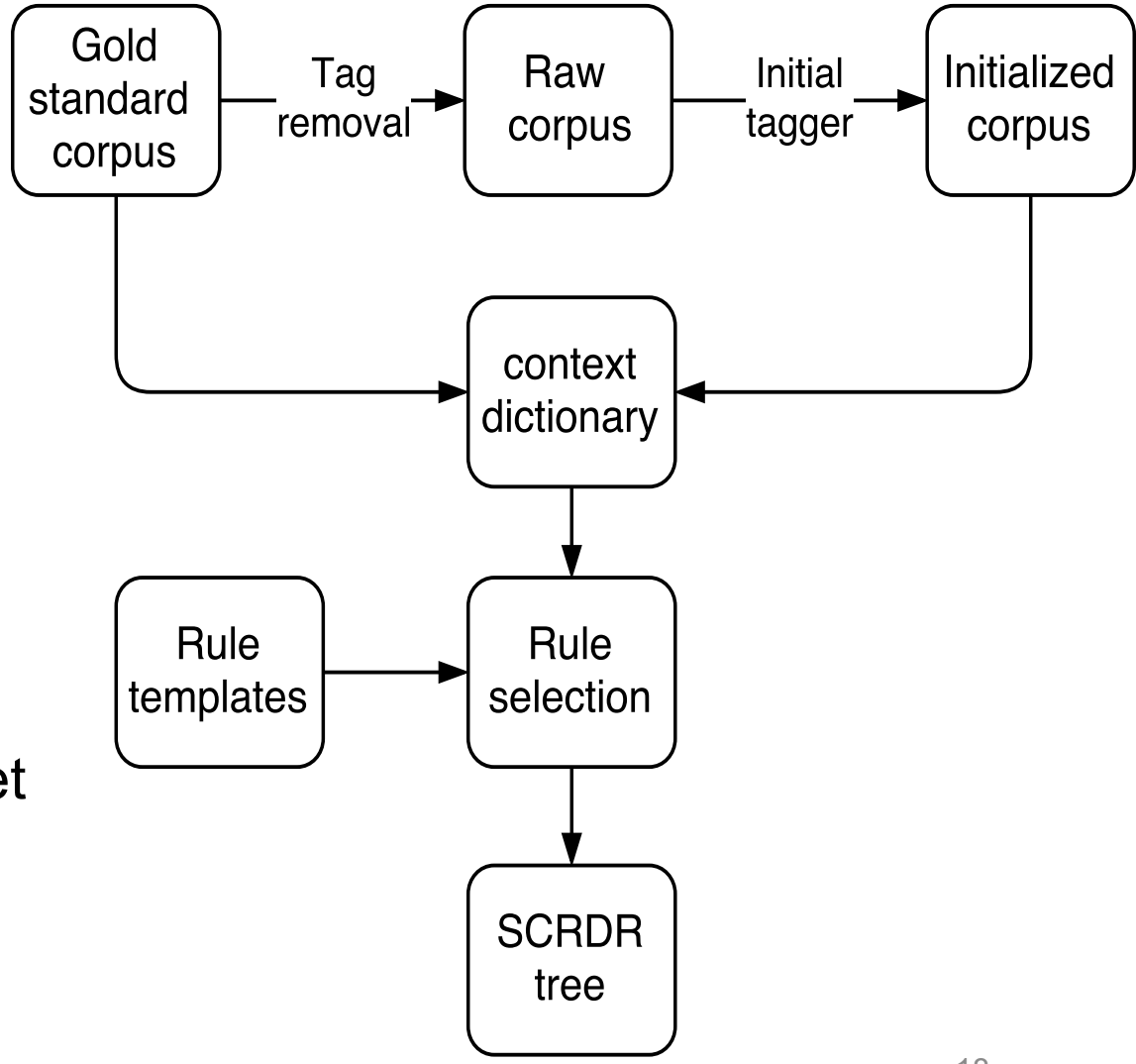
Rule-based framework for sequence labeling

- SCRDR tree initialization:



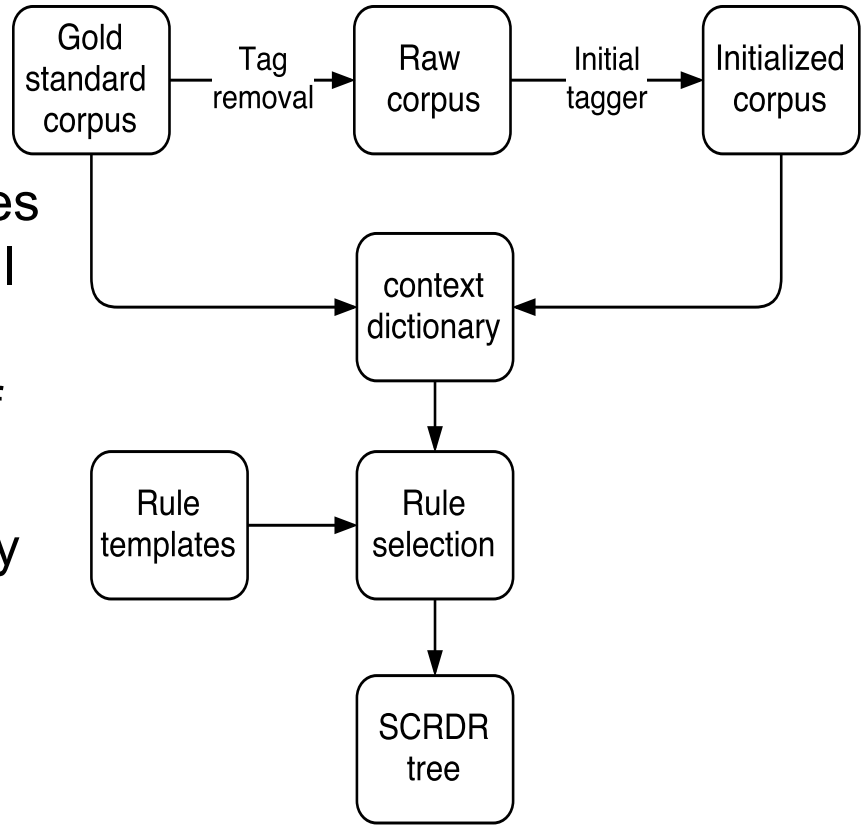
- At a current node the tree, we determine a subset of the context dictionary:
  - the node is the last fired node in the evaluation path for every tuple in the subset but the node returns an incorrect label

Example: given node (1), we have:
(as, IN, investors, NNS, **anticipate, VB**, a, DT, recovery, NN, te, ate, pate)



Rule-based framework for sequence labeling

- A a new exception rule must be added to the current tree to correct the errors given by the current node
  - The new exception rule is selected from all concrete rules which are generated by applying the rule templates to all tuples in the subset
  - The selected rule is associated with the highest value of the subtraction a – b:
    - a is the number of tuples in which each tuple not only satisfies the rule's condition but also gets a correct label given by the rule's conclusion
    - b is the number of tuples in which each tuple also satisfies the rule's condition but gets a wrong label given by the rule's conclusion
  - the selected rule's condition must not be satisfied by every tuple for which the current node already returns a correct label



Experimental results for POS tagging

- Tool: RDRPOStagger http://rdrpostagger.sourceforge.net/ (100+k words/second)
- Vietnamese: get the highest result at the POS tagging shared task at the second Vietnamese language and speech processing workshop 2013
- English: 96.86% using the standard evaluation setup on Penn Treebank
- German: Using the 10-fold cross validation evaluation scheme on the TIGER corpus

Model	**Accuracy**
TreeTagger	96.89
TnT	96.92
SVMTool	97.12
Stanford tagger	97.63
Apache UIMA Tagger	96.04
MarMot	97.85
$\text{RDRPOSTagger}_{+\text{TnT}}$	97.46

Experimental results for POS tagging

	Tr.T	#rules	EL	Tg.S
Bulgarian*	4	2,006	4	136K
Czech*	79	17,455	5	44K
Dutch*	53	6,468	5	87K
English	25	2,418	4	208K
French	19	1,376	4	207K
French*	11	2,953	5	215K
German	37	2,560	4	163K
German*	23	13,782	5	91K
Hindi	26	3,094	4	163K
Italian	4	1,189	4	275K
Lao	1	227	4	237K
Portuguese*	51	4,785	5	141K
Spanish*	6	1,699	4	221K
Swedish*	49	5,898	5	129K
Thai	7	1,405	4	264K
Vn (VTB)	8	765	4	216K
Vn (VLSP)	38	2,348	4	118K

Language	**RDRPOSTagger$_{+TnT}$**			**MarMoT**				
	Tagging accuracy			Tagging accuracy			Speed	
	Kno.	Unk.	All.	Kno.	Unk.	All.	TT	TS
Bulgarian*	96.82	70.27	94.12	96.92	76.72	**94.86**$^{+}$	9	4K
Czech*	93.24	67.92	91.70	94.74	75.84	**93.59**$^{+}$	130	2K
Dutch*	94.00	69.20	92.17	94.74	73.39	**93.17**$^{+}$	44	3K
English	97.17	86.19	96.86	97.47	89.39	**97.24**	5	16K
French	98.27	87.55	97.70	98.33	91.15	**97.93**	2	12K
French*	95.42	70.93	94.16	95.55	77.66	**94.62**$^{+}$	9	6K
German	98.13	89.43	97.46	98.30	92.54	**97.85**$^{+}$	5	9K
German*	87.65	62.05	85.66	90.61	69.13	**88.94**$^{+}$	32	3K
Hindi	—	—	96.21	—	—	**96.61**$^{+}$	3	16K
Italian	96.75	86.18	95.49	96.90	89.21	**95.98**$^{+}$	2	6K
Portuguese*	96.30	78.81	95.53	96.53	81.49	**95.86**$^{+}$	23	6K
Spanish*	99.05	84.13	98.26	99.08	86.86	**98.45**$^{+}$	8	8K
Swedish*	96.79	85.68	95.81	97.15	86.63	**96.22**$^{+}$	11	7K
Thai	95.03	81.10	94.21	95.42	86.99	**94.94**$^{+}$	2	12K
Vn (VTB)	94.15	59.39	92.95	94.37	69.89	**93.53**$^{+}$	1	16K
(VLSP)	94.16	68.14	93.63	94.52	75.36	**94.13**$^{+}$	3	21K

Extension for Vietnamese word segmentation

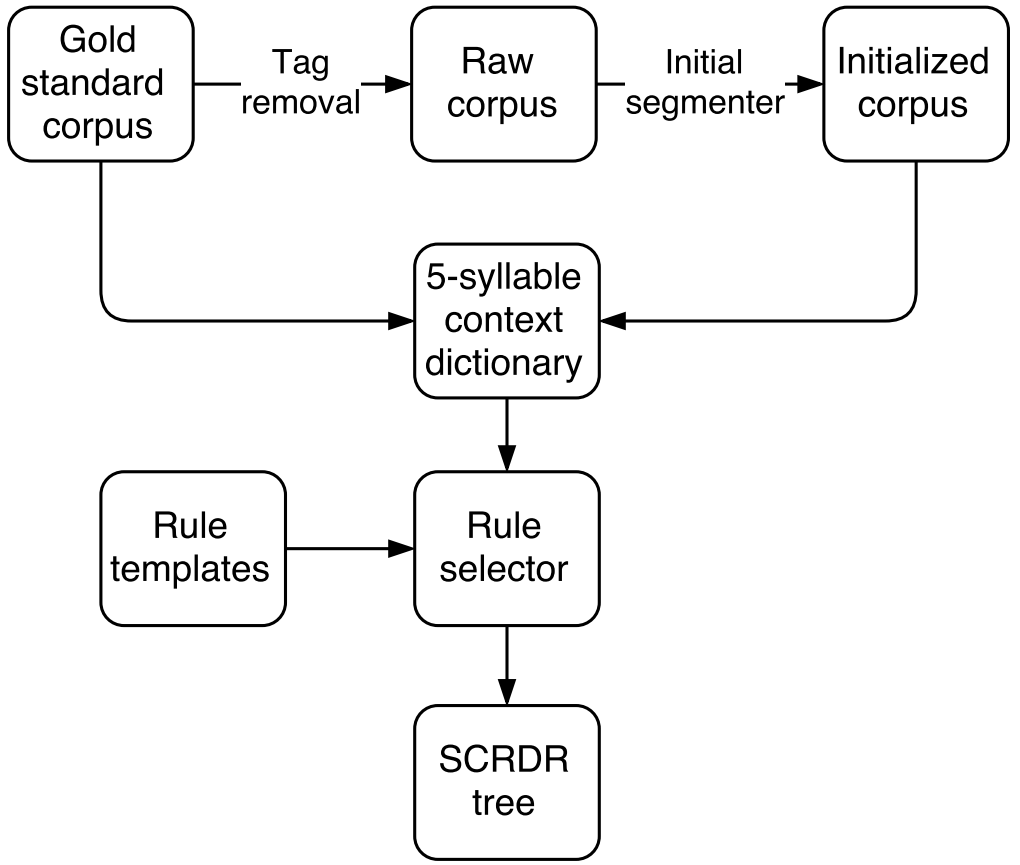
- We formalize the word segmentation problem as a sequence labeling task: each syllable is labeled by either segmentation tag B (Begin of a word) or I (Inside of a word)

  "thuế_thu_nhập cá_nhân"

  ($individual_{cá\_nhân}$ $income_{thu\_nhập}$ $tax_{thuế}$)

  => thuế/B thu/I nhập/I cá/B nhân/I

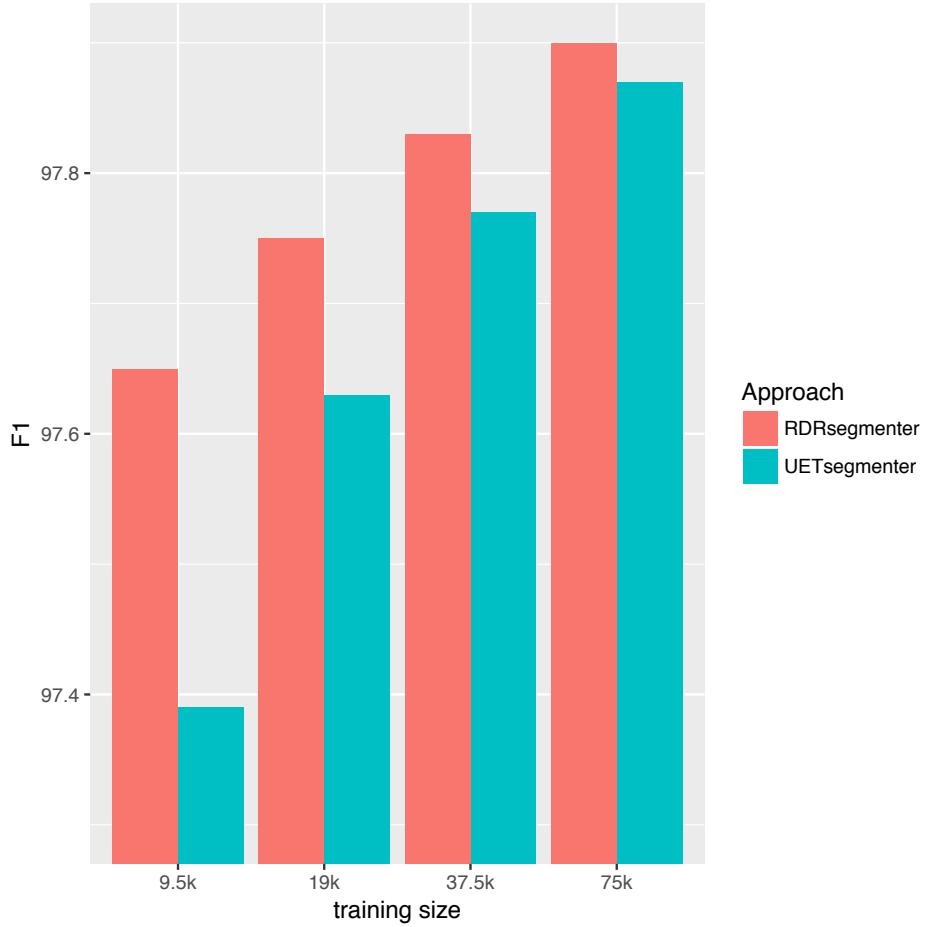
- Initial word segmenter is based on longest matching technique



Results for Vietnamese word segmentation

- Highest scores in terms of accuracy and performance speed (62k words/second)

Approach	Precision	Recall	F_1
vnTokenizer	96.98	97.69	97.33
JVnSegmenter-Maxent	96.60	97.40	97.00
JVnSegmenter-CRFs	96.63	97.49	97.06
DongDu	96.35	97.46	96.90
UETsegmenter	**97.51**	98.23	97.87
Our **RDRsegmenter**	97.45	**98.33**	**97.90**



Conclusions

- A rule-based framework for sequence labeling
- Fast and light-weight
- Competitive results