

APPENDIX A: IMPROVING POS TAGGING PHASE WITH ADJUSTING RULES

In this step, the tokens that are “Noun” and “Verb” are detected and adjusted with the correct POS tag. For doing so, patterns that take into account the position (index) in the sentence, POS tag of the token, POS tag of previous tokens and POS tag of next tokens are used. To extract these patterns, a study was performed with existing input projects (datasets): the tokens that are “Noun” and “Verb” for each sentence were detected and afterwards patterns (regular expression that matches the sequences of tags in a chunk) were extracted according to the position of the token in the sentence.

Fig. 1 illustrates a few rules for adjusting POS Tags. For instance, in the sentence “User types in the numbers of his PIN and presses the Enter button”, Stanford Core NLP POS tagger wrongly tagged the verbs “types” and “presses” as nouns/NNS. Then, an *adjusting rule* puts the right tags (verb-VBZ) because “types” and “presses” are words that are *NOUN_AND_VERB*, and they are between noun(NN)/conjunction(CC) and preposition(IN)/determiner(DT) words.

Wrong POS Tagging?	Rule for Adjusting	Adjusted POS Tagging
<p>User selects the type</p>	<p>token.POS = NN NNS; token ∈ NOUN_AND_VERB; Prev_Token = (DT PDT IN POS PRP\$ JJ.?) (IN (DT)? NN.?) THEN: token.confirmedNoun = TRUE;</p>	<p>User selects the type</p>
<p>Use case ends</p>	<p>token.POS = NN NNS; token.confirmedNoun ≠ TRUE; token ∈ NOUN_AND_VERB; Prev_Token = (^ . CC); Next_Token = (NN.?(NN.?) VB.?) IF token_{i+2}.POS = VB.? & token_{i+2}.lemma ∈ NOUN_AND_VERB THEN token.confirmedNoun = TRUE;</p>	<p>Use case ends</p>
<p>User types in the numbers of his PIN and presses the Enter button</p>	<p>token.POS = NN NNS; token ∈ NOUN_AND_VERB; token.confirmedNoun ≠ TRUE; Prev_Token ≠ CC; Next_Token = (DT PDT IN NN.?(PRP.?) RB.?(JJ.?) VB.?(CD)); THEN: token.POS = VBP VBZ</p>	<p>User types in the numbers of his PIN and presses the Enter button</p>
<p>Process bids</p>	<p>token.POS = NN NNS; token ∈ NOUN_AND_VERB; token.confirmedNoun = TRUE; Prev_Token = (^ . CC); Next_Tokens = (IN RP)? (NN.?(DT PDT JJ.?) VBD VBN RB.?(PRP.?) THEN: token.POS = VBP VBZ</p>	<p>Process bids</p>
<p>User register or delete transactions</p>	<p>token.POS = NN NNS; token ∈ NOUN_AND_VERB; token.confirmedNoun ≠ TRUE; Prev_Token = ((PDT DT)? (NN.?(JJ.?) VBD VBN)? NN.?) Next_Tokens = (\$ CC VB.?) THEN: token.POS = VBP VBZ</p>	<p>User register or delete transactions</p>
<p>Scenario finishes</p>	<p>token.POS = VB VBP VBZ; token ∈ NOUN_AND_VERB; Next_Tokens = (IN VB.?) THEN: token.POS = VBP VBZ</p>	<p>Scenario finishes</p>
<p>System displays list of possible criteria</p>	<p>token.POS = VB VBP VBZ; token ∈ NOUN_AND_VERB; Next_Tokens = (IN VB.?) THEN: token.POS = NN NNS</p>	<p>System displays list of possible criteria</p>
<p>Candidate fills the registration data forms</p>	<p>token.POS = VB VBP VBZ; token ∈ NOUN_AND_VERB; Prev_Token = ((VB VBZ VBP) (PDT DT)? (DT IN TO NN.?(JJ.?) VBD VBN)? NN.?) Next_Tokens = (\$ CC (^ (VB VBP VBZ))) THEN: token.POS = NN NNS</p>	<p>Candidate fills the registration data forms</p>

Fig. 11. Rules for Adjusting NOUN and VERB POS tags.

Tables 1, 2 and 3 detail the rules to adjust the accuracy of POS tagging phase adding a second phase. These rules are presented in GIVEN-WHEN-THEN format and they are ordered. To adjust a token of a given sentence, the approach sequentially searches for some token properties in the GIVEN (Antecedent) part of each rule against the properties of the given token. In case of *antecedent* is satisfied, the WHEN part of the rule is activated; the approach sequentially searches for all POS tags in the PREVIOUS (tokens) and NEXT (tokens) part of each rule against the POS tags of the given sentence. In case of matches, the THEN (Consequent) part of the rule is used to determine the correct POS tag of the given token.

In Tables 1, 2 and 3, $tokens = \{token_i, token_{i+1}, \dots, token_{n-1}, token_n\}$ is the set of tokens of a given sentence and $i = 1 \rightarrow$ length of tokens, $token_i = \{index, word, POS, lemma, nounAndVerb, confirmedNoun, confirmedVerb, ConfirmAdjective\}$ is a token and its properties, *NOUN_AND_VERB* is a set containing the words that are both *noun* and *verb*, $()$ is used for grouping, $|$ stands for “OR”, $?$ matches the preceding character 0 or 1 time, $+$ matches the preceding character 1 or more time, $*$ matches the preceding character 0 or more time, $^$ matches any single character except the newline character, $\$$ matches the end of the input, $^$ matches the beginning of the input (When used in a set pattern $([^abc])$, it negates the set; match anything not enclosed in the brackets).

TABLE 1
RULES FOR ADJUSTING POS TAGGING PHASE – ADJUST NOUNS.

Rule #	Description	GIVEN (Antecedent)	WHEN		THEN (Adjust Token)	Example
			Previous Tokens (contains POS tags:)	Next Tokens (contains POS tags:)		

PTR1	Check that a 'Noun' is effectively a 'Noun'. Prepositions are most commonly followed by a 'Noun' phrase or 'Pronoun'	$token_i.lemma \in NOUN_AND_VE$ RB $token_i.POS = NN \mid NNS$	$(DT \mid PDT \mid IN \mid POS \mid PRP\$ \mid JJ.?) \mid (IN \mid (DT)? \mid NN.?)$		$token_{i+2}.confirmedNoun = TRUE;$	The/DT Broker/NNP System/NNP displays/NNS the/DT count/NN of/IN the/DT customer/NN User/NN selects/VBZ the/DT type/NN User/NN selects/VBZ the/DT type/NN and/CC localization/NN of/IN the/DT output/NN file/NN with/IN report/NN
PTR2	Check that a 'Noun' is effectively a 'Noun'. The 'Noun' position is the first or after a coordinating conjunction (CC)	$token_i.confirmedNo_{un} = FALSE$ $token_i.lemma \in NOUN_AND_VE$ RB $token_i.POS = NN \mid NNS$	$(^ \wedge \mid .* CC)$	$(NN.?(NN.?(\mid VB.?)$	IF $token_{i+2}.POS = VB.?$ & $token_{i+2}.lemma \in NOUN_AND_VERB$ THEN $token_{i+2}.confirmedNoun = TRUE;$	use/NN case/NN ends/VBZ download/NN system/NN finishes/NN
PTR3	Check that a 'Noun' between a Determiner (or Preposition or Noun or Adverb) and a 'TO' is effectively a 'Noun'	$token_i.lemma \in NOUN_AND_VE$ RB $token_i.POS = NN \mid NNS$	$((VB \mid VBP \mid VBZ \mid VBD \mid VBN) (DT \mid PDT \mid IN \mid NN.?(RB) +)$	$(TO (DT \mid PDT \mid IN \mid NN.?(PRP\$ \mid JJ.?)$	$token_{i+2}.confirmedNoun = TRUE;$	System/NNP sends/VBZ a/DT registration/NN request/NN to/TO the/DT server/NN
PTR4	Check that a 'Noun' followed by a gerund verb is effectively a 'Noun'	$token_i.lemma \in NOUN_AND_VE$ RB $token_i.POS = NN \mid NNS$		VBG	$token_{i+2}.confirmedNoun = TRUE;$	Post/NN containing/VBG
PTR5	Check that a 'Noun' is effectively a 'Noun'. The 'Noun' is preceded by a Verb + Determiner + Noun or Adjective	$token_i.confirmedNo_{un} = FALSE$ $token_i.lemma \in NOUN_AND_VE$ RB $token_i.POS = NN \mid NNS$	$((VB \mid VBZ \mid VBP \mid VBD \mid VBN) (PDT \mid DT)? (NN.?(JJ.?)^*)$		IF $token_{i+1}.lemma \notin NOUN_AND_VERB$ THEN $token_{i+1}.confirmedNoun = TRUE;$	System/NNP presents/VBZ a/DT registration/NN data/NN form/NN and/CC asks/VBZ to/TO enter/VB the/DT registration/NN data/NN Candidate/NNP fills/VBZ the/DT registration/NN data/NN form/NN and/CC submits/VBZ the/DT registration/NN data/NN form/NN
PTR6	Check that a 'Noun' is a 'Verb'. The 'Noun' position is after a token, which is not a coordinating conjunction (CC)	$token_i.confirmedNo_{un} = FALSE$ $token_i.lemma \in NOUN_AND_VE$ RB $token_i.POS = NN \mid NNS$	$[^ \wedge (CC)]$	$(DT \mid PDT \mid IN \mid NN.?(PRP.?(RB.?(JJ.?(VB.?(CD)$	IF $token_{i+1}.lemma \notin NOUN_AND_VERB$ & $token_{i+1}.POS \neq VB.?$ THEN $token_{i+1}.confirmedVerb = TRUE;$ $token_{i+1}.POS = VB.?$	System/NN displays/NNS the/DT welcome/JJ interface/NN System/NNP checks/VBZ if/IN a/DT group/NN with/IN the/DT given/VBN name/NN has/VBZ not/RB been/VBN already/RB defined/VBN and/CC if/IN so/RB inserts/NNS the/DT name/NN of/IN a/DT new/JJ group/NN into/IN a/DT database/NN
PTR7	Check that a 'Noun' is a 'Verb'. The 'Noun' position is the first or after a coordinating conjunction (CC)	$token_i.confirmedNo_{un} = FALSE$ $token_i.lemma \in NOUN_AND_VE$ RB $token_i.POS = NN \mid NNS$	$(^ \wedge \mid .* CC)$	$(IN \mid RP)? (NN.?(DT \mid PDT \mid JJ.?(VBD \mid VBN \mid RB.?(PRP.?)$	$token_i.POS = VB$	Process/NN bids/NNS Show/NN alert/JJ message/NN Request/NN for/IN licence/NN User/NN types/NNS in/IN the/DT numbers/NNS of/IN his/PRP\$ PIN/NN and/CC presses/NNS the/DT Enter/VBP button/NN
PTR8	Check that a 'Noun' is a 'Verb'. The 'Noun' position is the first followed by a preposition or particle, or TO.	$token_i.confirmedNo_{un} = FALSE$ $token_i.lemma \in NOUN_AND_VE$ RB $token_i.POS = NN \mid NNS$	$^$	$(((IN \mid RP)?\$) \mid ((IN \mid RP)? TO))$	$token_i.POS = VB$	Search/NN Log/NN in/IN Log/NN in/IN to/TO the/DT system/NN
PTR9	Check that a 'Noun' is a 'Verb'. The 'Noun' position is the last or before a coordinating conjunction (CC)	$token_i.confirmedNo_{un} = FALSE$ $token_i.lemma \in NOUN_AND_VE$ RB $token_i.POS = NN \mid NNS$	$((PDT \mid DT)? (NN.?(JJ.?(VBD \mid VBN)^* \mid NN.?)$	$(\$ \mid CC VB.?)$	IF $token_{i+1}.lemma \notin NOUN_AND_VERB$ THEN $token_{i+1}.confirmedVerb = TRUE;$ $token_{i+1}.POS = VB.?$	User/NN register/NN or/CC delete/VBP transactions/NNS Scenario/NNP finishes/NNS User/NN ends/NNS the/DT scenario/NN finishes/NNS logged/VBD user/NN ends/NNS

						The/DT broker/NN system/NN finishes/NNS The/DT atm/NN system/NN ends/NNS
--	--	--	--	--	--	---

TABLE 2
RULES FOR ADJUSTING POS TAGGING PHASE – ADJUST VERBS.

Rule #	Description	GIVEN (Antecedent)	WHEN		THEN (Adjust Token)	Example
			Previous Tokens (contains POS tags:)	Next Tokens (contains POS tags:)		
PTR10	Check that a 'Verb' is effectively a 'Verb'. Prepositions are most commonly followed by a 'Noun' phrase or 'Pronoun'	$token_i.lemma \in \text{NOUN_AND_VE}$ RB $token_i.POS = \text{VB} \mid \text{VBP} \mid \text{VBZ}$	(DT PDT IN POS PRP\$ JJ.?)		$token_{i+1}.confirmedNoun = \text{TRUE};$ $token_i.POS = \text{NN.}?$	User/NN types/NNS in/IN the/DT numbers/NNS of/IN his/PRP\$ PIN/NN and/CC presses/NNS the/DT Enter/VBP button/NN
PTR11	Check that a 'Verb' is effectively a 'Verb'. The 'Verb' position is after a token, which is not a coordinating conjunction (CC)	$token_{i-1}.confirmedVerb = \text{FALSE}$ $token_i.lemma \in \text{NOUN_AND_VE}$ RB $token_i.POS = \text{VB} \mid \text{VBP} \mid \text{VBZ}$	[^ (CC)]	(DT PDT IN NN.? PRP.? RB.? JJ.? VB.?)	IF $token_{i-1}.token_{i-1}.lemma \notin \text{NOUN_AND_VERB}$ THEN $token_{i-1}.confirmedVerb = \text{TRUE};$	System/NNP displays/VBZ list/VB of/IN possible/JJ criteria/NNS
PTR12	Check that a 'Verb' is effectively a 'Verb'. The 'Verb' is preceded by a Noun and followed by 'TO' + Verb	$token_{i-1}.confirmedVerb = \text{FALSE}$ $token_{i-2}.lemma \in \text{NOUN_AND_VE}$ RB $token_i.POS = \text{VB} \mid \text{VBP} \mid \text{VBZ}$	(PRP NN.?)	(TO (VB.? NN.?)	IF $token_{i-1}.lemma \notin \text{NOUN_AND_VERB}$ & $token_{i-2}.lemma \in \text{NOUN_AND_VERB}$ THEN $token_{i-1}.confirmedVerb = \text{TRUE};$	User/NN proceeds/VBZ to/TO print/VB
PTR13	Check that a 'Verb' followed by "OF" or TO_BE or TO_HAVE is a 'Noun'	$token_{i+1}.confirmedVerb = \text{FALSE}$ $token_i.lemma \in \text{NOUN_AND_VE}$ RB $token_i.POS = \text{VB} \mid \text{VBP} \mid \text{VBZ} \mid \text{VBD} \mid \text{VBN}$		(IN VB.?)	IF $token_{i+1}.word = \text{'of'}$ $token_{i+1}.lemma = \text{'be'}$ $token_{i+1}.lemma = \text{'have'}$ THEN $token_i.POS = \text{NN.}?$	System/NNP displays/VBZ a/DT tree/NN view/VB of/IN available/JJ groups/NNS and/CC channels/NNS and/CC marks/VBZ it/PRP system/NN queries/VBZ the/DT database/NN for/IN news/NN messages/NNS ./, whose/WRP\$ expiry/JJ date/NN and/CC time/VB have/VBP passed/VBN
PTR14	Check that a 'Verb' is a 'Noun'. The 'Verb' position is the last or before a coordinating conjunction (CC)	$token_{i+1}.confirmedVerb = \text{FALSE}$ $token_i.lemma \in \text{NOUN_AND_VE}$ RB $token_i.POS = \text{VB} \mid \text{VBP} \mid \text{VBZ}$	((VB VBZ VBP) (PDT DT)? (DT IN TO NN.? JJ.? VBD VBN)* NN.?)	(\$ CC [^ (VB VBP VBZ)])	$token_i.POS = \text{NN.}?$	User/NN fills/VBZ all/DT required/VBD personal/JJ <u>client/NN data/NNS</u> forms/VBZ System/NNP presents/VBZ a/DT registration/NN data/NN form/NN and/CC asks/VBZ to/TO enter/VB the/DT registration/NN data/NNS

TABLE 3
RULES FOR ADJUSTING POS TAGGING PHASE – ADJUST PREPOSITIONS AND ADJECTIVES.

Rule #	Description	GIVEN (Antecedent)	WHEN		THEN (Adjust Token)	Example
			Previous Tokens (contains POS tags:)	Next Tokens (contains POS tags:)		
PTR15	Check that a 'Preposition' is a 'Verb'. The 'Preposition' position is the first or after a coordinating conjunction (CC)	$token.lemma \in NOUN_AND_VE$ RB $token.POS = IN$	([^] CC)	(NN.? DT PDT JJ.?)	$token.POS = VB$	<u>Post/IN</u> a/DT group/NN message/NN
PTR16	Check that a 'Preposition' is a 'Verb'. The 'Preposition' position is after a token, which is not a coordinating conjunction (CC)	$token.lemma \in NOUN_AND_VE$ RB $token.POS = IN$	(PRP NN.? RB.?)	(NN.? PRP.? WDT WP.? WRB)	$token.POS = VBP$	User/NN <u>like/IN</u> it/PRP
PTR17	Check that a 'Adjective' is effectively a 'Adjective'. Prepositions are most commonly followed by a 'Noun' phrase or 'Pronoun' or Adjective	$token.lemma \in NOUN_AND_VE$ RB $token.POS = JJ$	(DT IN POS PRP\$ JJ.?)		$token.confirmedAdjective = TRUE;$	System/NN <u>displays/NNS</u> the/DT welcome/JJ interface/NN
PTR18	Check that an 'Adjective' is a 'Verb'. Modifiers are most commonly followed by adjectives	$token.confirmedAdjective = FALSE;$ $token.lemma \in NOUN_AND_VE$ RB $token.POS = JJ$	([^] (VB.?) [^])	(DT PDT IN NN.? PRP.? JJ.? RB.? VBD)	IF $token.i.lemma \notin NOUN_AND_VERB$ THEN $token.POS = VB.?$	User/NN <u>select/JJ</u> option/NN to/TO adding/VBG new/JJ clients/NNS User/NN <u>select/JJ</u> option/NN for/IN searching/VBG User/NN <u>select/delete/JJ</u> transactions/NNS
PTR19	Check that a 'Verb' with -ing and -ed is effectively an 'Adjective'. Adjectives are most commonly preceded by a Determiners.	$token.POS = VBD VBN VBG$	(DT PDT JJ.? IN)		IF $token.i.POS \neq IN$ & $token.POS \neq VBG$ THEN $token.POS = JJ;$ $token.confirmedAdjective = TRUE;$	Server/NN sends/VBZ separate/JJ news/NN messages/NNS from/IN all/DT <u>subscribed/VBD</u> channels/NNS System/NNP receives/VBZ a/DT RSS-like/JJ <u>formatted/VBD</u> news/NN file/NN
PTR20	Check that a 'Verb' with -ing and -ed is an 'Adjective'. The 'Adjective' is preceded by a Verb (+ Determiner + Noun or Adjective)	$token.confirmedAdjective = FALSE;$ $token.POS = VBD VBN VBG$	((VB VBZ VBP) (DT PDT)? (NN.? JJ.? RB.?)*)	NN.?	IF ($token.POS = VBN VBD$ & $token.i.lemma = 'have'$) ($token.POS = VBG$ & $token.i.lemma = 'be'$) THEN $token.confirmedVerb = TRUE;$ ELSE $token.POS = JJ;$ $token.confirmedAdjective = TRUE;$	The/DT Broker/NNP System/NNP <u>displays/NNS</u> payment/NN <u>denied/VBD</u> page/NN The/DT Broker/NNP System/NNP <u>displays/VBZ</u> a/DT payment/NN <u>denied/VBD</u> page/NN The/DT Broker/NNP System/NNP <u>displays/VBZ</u> the/DT online/JJ payment/NN system/NN <u>denied/VBD</u> page/NN