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.

Table 16, 17 and 18 show 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 matchs, the THEN (Consequent) part of the rule is used to determine the correct POS tag of the given token.

In Table 16, 17 and 18, *tokens* = {$token_i$, $token_{i+1}$, …, $token_{n-1}$, $token_n$} is the set of tokens of a given sentence and i = 1 → 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 containg 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 16
RULES FOR ADJUSTING POS TAGGING PHASE – ADJUNST 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_VERB $token_i.POS$ = NN | NNS	(DT | PDT | IN | POS | PRP$ | JJ.?) | (IN (DT)? NN.?)		$token_i.confirmedNoun$ = 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)	$token_i.confirmedNoun$ = FALSE $token_i.lemma \in$ NOUN_AND_VERB $token_i.POS$ = NN | NNS	(^ | .* CC)	(NN.? (NN.? | VB.?))	**IF** $token_{i+2}.POS$ = VB.? & $token_{i+2}.lemma \in$ NOUN_AND_VERB **THEN** $token_i.confirmedNoun$ = 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_VERB $token_i.POS$ = NN | NNS	((VB | VBP | VBZ | VBD | VBN) (DT | PDT | IN | NN.? | RB) +)	(TO (DT | PDT | IN | NN.? | PRP$ | JJ.?))	$token_i.confirmedNoun$ = 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_VERB $token_i.POS$ = NN | NNS		VBG	$token_i.confirmedNoun$ = 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	$token_i.confirmedNoun$ = FALSE $token_i.lemma \in$ NOUN_AND_VERB $token_i.POS$ = NN | NNS	((VB | VBZ | VBP | VBD | VBN) (PDT | DT)? (NN.? | JJ.?)*)		**IF** $token_{i-1}.lemma \notin$ NOUN_AND_VERB **THEN** $token_i.confirmedNoun$ = 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/NNS 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)	*$token_i$.confirmedNoun* = FALSE
		$token_i$.lemma ∈ NOUN_AND_VERB
		$token_i$.POS = NN | NNS
			[^ (CC)]	(DT | PDT | IN | NN.? | PRP.? | RB.? | JJ.? | VB.? | CD)	**IF** *$token_{i-1}$.lemma* ∉ NOUN_AND_VERB & *$token_{i-1}$.POS* != VB.?
					THEN
					~~*$token_i$.confirmedVerb* = TRUE;~~
					$token_i$.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)	*$token_i$.confirmedNoun* = FALSE *$token_i$.lemma* ∈ NOUN_AND_VERB *$token_i$.POS* = NN | NNS	(^ | .* CC)	(IN | RP)? (NN.? | DT | PDT | JJ.? | VBD | VBN | 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.	*$token_i$.confirmedNoun* = FALSE *$token_i$.lemma* ∈ NOUN_AND_VERB *$token_i$.POS* = NN | NNS	^	(((IN | RP)?$) | ((IN | RP)? TO))	*$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)	*$token_i$.confirmedNoun* = FALSE
		$token_i$.lemma ∈ NOUN_AND_VERB
		$token_i$.POS = NN | NNS
			((PDT | DT)? (NN.? | JJ.? | VBD | VBN)* NN.?)	($ | CC VB.?)	**IF** *$token_{i-1}$.lemma* ∉ NOUN_AND_VERB
					THEN
					~~*$token_i$.confirmedVerb* = TRUE;~~
					$token_i$.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 17
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* ∈ NOUN_AND_VERB *token$_i$.POS* = VB | VBP | VBZ	(DT | PDT | IN | POS | PRP$ | JJ.?)		*token$_i$.confirmedNoun* = TRUE; *token$_i$*.POS = 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$_i$.confirmedVerb*= FALSE~~ *token$_i$.lemma* ∈ NOUN_AND_VERB *token$_i$.POS* = VB | VBP | VBZ	[^ (CC)]	(DT | PDT | IN | NN.? | PRP.? | RB.? | JJ.? | VB.?)	**IF** *token$_{i-1}$ token$_{i-1}$.lemma* ∉ NOUN_AND_VERB **THEN** *token$_i$.confirmedVerb* = 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$_i$.confirmedVerb*= FALSE~~ *token$_i$.lemma* ∈ NOUN_AND_VERB *token$_i$.POS* = VB | VBP | VBZ	(PRP | NN.?)	(TO (VB.? | NN.?))	**IF** *token$_{i-1}$.lemma* ∉ NOUN_AND_VERB & *token$_{i+2}$.lemma* ∈ NOUN_AND_VERB **THEN** *token$_i$.confirmedVerb* = 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$_i$.confirmedVerb* = FALSE~~ *token$_i$.lemma* ∈ NOUN_AND_VERB *token$_i$.POS* = VB | VBP | VBZ | VBD | VBN		(IN | VB.?)	**IF** *token$_{i+1}$.word* == 'of' | *token$_{i+1}$.lemma* == 'be' | *token$_{i+1}$.lemma* == 'have' **THEN** *token$_i$*.POS = 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/WP$ 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$_i$.confirmedVerb* = FALSE~~ *token$_i$.lemma* ∈ NOUN_AND_VERB *token$_i$.POS* = VB | VBP | VBZ	((VB | VBZ | VBP) (PDT | DT)? (DT | IN | TO | NN.? | JJ.? | VBD | VBN)* NN.?)	($ | CC [^ (VB | VBP | VBZ)])	*token$_i$*.POS = NN.?	User/NN fills/VBZ all/DT required/VBD personal/JJ client/NN data/NNS **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 18
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)~~	$token_i$.*lemma* ∈ NOUN_AND_VERB $token_i$.*POS* = IN	(^ | CC)	(NN.? | DT | PDT | JJ.?)	$token_i$.POS = VB	**Post/IN** 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)~~	$token_i$.*lemma* ∈ NOUN_AND_VERB $token_i$.*POS* = IN	(PRP | NN.? | RB.?)	(NN.? | PRP.? | WDT | WP.? | WRB)	$token_i$.POS = VBP	User/NN **like/IN** it/PRP
PTR17	Check that a 'Adjective' is effectively a 'Adjective'. Prepositions are most commonly followed by a 'Noun' phrase or 'Pronoun' or Adjective	$token_i$.*lemma* ∈ NOUN_AND_VERB $token_i$.*POS* = JJ	(DT | IN | POS | PRP$ | JJ.?)		$token_i$.*confirmedAdjective* = TRUE;	System/NN **displays/NNS** the/DT welcome/JJ interface/NN

PTR18	Check that an 'Adjective' is a 'Verb'.
	Modifiers are most commonly followed by adjectives
		$token_i$.*confirmedAdjective* = FALSE;
		$token_i$.*lemma* ∈ NOUN_AND_VERB
		$token_i$.*POS* = JJ
			([^ (VB.?] | ^)	(DT | PDT | IN | NN.? | PRP.? | JJ.? | RB.? | VBD)	**IF** $token_{i-1}$.*lemma* ∉ NOUN_AND_VERB
					THEN
					$token_i$.POS = VB.?
						User/NN **select/JJ** option/NN to/TO adding/VBG new/JJ clients/NNS
						User/NN **select/JJ** option/NN for/IN searching/VBG
						User/NN **select/delete/JJ** transactions/NNS

PTR19	Check that a 'Verb' with -ing and -ed is effectively an 'Adjective'.
	Adjectives are most commonly preceded by a Determiners.
		$token_i$.*POS* = VBD | VBN | VBG	(DT | PDT | JJ.? | IN)		**IF** $token_{i-1}$.*POS* ≠ IN & $token_i$.*POS* ≠ VBG
					THEN $token_i$.POS = JJ;
					$token_i$.*confirmedAdjective* = TRUE;
						Server/NN sends/VBZ separate/JJ news/NN messages/NNS from/IN all/DT **subscribed/VBD** channels/NNS
						System/NNP receives/VBZ a/DT RSS-like/JJ **formatted/VBD** 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)	$token_i$.*confirmedAdjective* = FALSE;
		$token_i$.*POS* = VBD | VBN | VBG
			((VB | VBZ | VBP) (DT | PDT)? (NN.? | JJ.? | RB.?)*)	NN.?	**IF** ($token_i$.*POS* == VBN | VBD & $token_{i-1}$.*lemma* == 'have') | ($token_i$.*POS* == VBG & $token_{i-1}$.*lemma* == 'be')
					THEN
					$token_i$.*confirmedVerb* = TRUE;
					ELSE
					$token_i$.POS = JJ;
					$token_i$.*confirmedAdjective* = TRUE;
						The/DT Broker/NNP System/NNP displays/NNS payment/NN **denied/VBD** page/NN
						The/DT Broker/NNP System/NNP displays/VBZ a/DT payment/NN **denied/VBD** page/NN
						The/DT Broker/NNP System/NNP **displays/VBZ** the/DT online/JJ payment/NN system/NN **denied/VBD** page/NN