ID	Elements of Interest	Breakdowns	input	Typed dependencies	Extraction output	Versions
R1	**Subject**			➔nsubj(X, **Y**) ➔nsubjpass (X, **Y**)	Subject = Y	
R2	**Verb**			➔ nsubj(X, **Y**) ➔ nsubjpass (X, **Y**) ➔ obj(X, **Y**) ➔ dobj(X, **Y**) ➔ idobj(X, **Y)**	Verb = X	
R3	**Verb Arguments**		X: is a verb Z: is not Keyword	➔ obj(X, **Y**) ➔ dobj(X, **Y**) ➔ idobj(X, **Y**) ➔ case(Y, Z), nmod(X, Y)	Object = Y Preposition = Z	One ore more object. Each with optional preceded preposition
R4	**Predicate**	Subj		R1		subj+verb+verbArgs
		Verb		R2		subj+verb
		verbArgs		R3		verb+ verbArgs
				Core-segment **Signature: [Head][Subj][Verb][VerbArgs]**		
R5	**dependent-Clauses {Condition,Trigger, ReqScope}**	Head	X:{if|when|while|before|until|after} Y: verb	Head: keyword ➔mark(Y, X) ➔advmod(Z, X) ➔case(Z, X),nmod(Y, Z)	Head= X	Head Keywords resulted from the unification process • Condition: {if} • Trigger: {when} • ReqScope: {while, before, until, after}
		Predicate		R4		
R6	**Independent clauses {Action,factual-rule}**	Predicate		R4		
				Hidden-constraint **Signature: [Rel-Noun][Rel-Head][Subj][Verb][VerbArgs]**		
R7	**Rel clauses {hiddenConstraint}**	rel Noun rel head Property	X: {whose|that} Y: is verb	➔dep(Y, X), nsubj(Y,V), dep(Z, Y) ➔nsubj(Y, X), acl:relcl(Z, Y)	relative head = X relative noun = Z Property = V	Property related - relative clause • the signal whose index is larger than 2 shall be set to 5. Noun related - relative clause • the signal that is larger than 2 shall be set to 5.
		Verb		R2		
		verbArgs		R3		
				Time **Signature: [Preposition][Quantification][Value][Unit]**		
R8	**Time**	Preposition+ quantification + value + unit				
		Preposition value unit	X: is not keyword Y:{every}	➔ case(U, X), nummod(U, V) ➔ dep(V, P1), mwe(P1, P2), nummod(U, V) -> X:P1+P2 ➔det(U, Y), nummod(U, V). ➔ advmod(U, Y), nummod(U, V).	Preposition= X|Y Value = V Unit= U	
		Quantification	V: Value U: Unit	➔ case(JJS, at), nmod:npmod(V, JJS) ➔ advmod(V, JJR), mwe(Adj, than) ➔ advmod(V, JJR), mwe(Adj, than), cc(V, or), advmod(V, equal) ➔ cc(U, or), conj(U, JJR) ➔ cc(JJR, or), nummod(U, JJR)	Quantification = (at+ JJS)| (JJR + than)| (JJR + than + or + equal)| (or+JJR)	Quantification: • at+ JJS ➔at least • JJR + than ➔ less than • JJR + than + or + equal ➔ less than or equal • Or +JJR ➔ or more (e.g., 3 or more)