Stanford CoreNLP Enhanced Dependencies Parser for SVO Extraction

One of the NLP Suite SVO extractors is based on Stanford CoreNLP enhanced dependencies (**EnhancedPlusPlusDependencies**) parser. The NLP Suite algorithm processes the json output file of the parser via a set of specialized functions that check for negative forms, passive forms, and many other linguistic forms that may otherwise lead to invalid SVOs.

Standard parsing

In Stanford CoreNLP, dependency parsing module builds a tree structure of tokens from the input sentence, which represents the syntactic dependency relations between tokens(https://stanfordnlp.github.io/CoreNLP/depparse.html). Each token in the output json of Stanford CoreNLP's dependent parsing module has these two properties: "dep" / "deprel"(the dependency relation between this token and its syntactic head) and "governor"(The id of the syntactic head of this token in the sentence) (https://stanfordnlp.github.io/stanfordnlp/depparse.html). Based on the output of Stanford CoreNLP, a new json is generated by traverse tree structure built by the dependency parsing module. That contains a token's major annotated information (https://stanfordnlp.github.io/CoreNLP/annotators.html), as well as a govern dictionary in which keys are the dep of other tokens whose syntactic head is that token, and values are the index of these deps in the sentence.

```
6: {'deprel': 'ROOT',
    'govern_dict': {'amod': 5,
                    'appos': 15,
                    'conj': 9,
                    'cop': 3,
                    'det': 4,
                    'nsubj': 2,
                    'punct': [7, 13, 25]},
    'lemma': 'length',
    'ner': 'O',
    'pos': 'NN',
    'word': 'length'},
7: {'deprel': 'punct',
    'govern_dict': {},
    'lemma': ',',
    'ner': 'O',
    'pos': ',',
    'word': ','},
8: {'deprel': 'nmod:poss',
    'govern_dict': {},
    'lemma': 'they',
    'ner': 'O',
    'pos': 'PRP$',
    'word': 'their'},
```

One verb, as a syntactic head, usually governs its subject (dep: “nsubj”) and object (dep: “obj”). For verbs, whose postag contains “VB” (https://stanfordnlp.github.io/CoreNLP/pos.html), if in the govern dictionary there’s either a subject or an object, they collocation will be extracted as an SVO collocation (if there’s a subject but without object, the object will be blank; if there’s only object, the default subject is “Someone?”).

Passive sentences

The SVO script converts passive sentences to active ones. In the dependency parser (https://nlp.stanford.edu/software/nndep.html) of Stanford CoreNLP, the object in passive sentence would be tagged as “nsubj:pass”, and the POS tag (https://stanfordnlp.github.io/CoreNLP/pos.html) of the verb would usually be “VBN” (although sometimes the Stanford CoreNLP incorrectly tags a passive tense as an adjective). When the passive subject (the agent) of the sentence is not available, the script will insert the default unknown subject as **Someone?**

Negation

The meaning of a sentence captured by the simple SVO structure changes drastically in the presence of negation (e.g., “I will give you a present” vs. “I will not give you a present”). The NLP Suite SVO algorithm captures a variety of negation forms (“no”, “not”, “n’t”, “seldom”, “never”, “hardly”, “neither”, “nor”, “without”).

By recursion with deep-first search through a token’s adverbial modifier (dep: “advmod”), determiner (dep: “det”), auxiliary (dep: “aux”) (https://universaldependencies.org/docs/u/dep/aux_.html), and coordinating conjunction (dep: “cc” and “cc:preconj”) (https://universaldependencies.org/docs/u/dep/), the negation would be

detected if the searched token is in a list of words with meaning of negation. If any of the subject, verb, or object in an SVO collocation contains negation, the result of the negation detection of this collocation would be TRUE. If multiple verbs are conjuncts that are connected by “or” and there’s negation detected in the first verb, the result of the negation detection of the following verbs will be set as TRUE.

Predicate Nominative

From https://www.thesaurus.com/e/grammar/predicate-nominative-vs-predicate-adjectives/ we read: “In general, a predicate completes a sentence by providing information about what the subject is or does. The subject of a sentence is who or what is doing the action. The predicate explains the action. There’s often a linking verb (like is or became) in between the two. A predicate nominative is a noun that completes the linking verb in a sentence. Predicate adjectives complete the linking verb by describing the subject of a sentence.”

“For example, ‘Ben is a fireman’ can read ‘Ben equals a fireman’ without changing the point. In this example, fireman is the predicate nominative.”

“For example, in ‘Jack is handsome,’ Jack is the subject, and handsome is the predicate adjective.”

In other words, when the lemma (https://stanfordnlp.github.io/CoreNLP/lemma.html) of a verb is “be”, it is usually not followed by an object but by a predicate nominative. In the sentence “He is a traitor”, “traitor” is a predicate nominative. In this situation, if the extraction starts from the govern dictionary of the verb “is”, that collocation will be omitted, since neither “He” nor “traitor” is governed by the verb. In a subject-verb-predicate collocation, the object is the syntactic head, which governs the subject (dep: “nsubj”) and the link verb (dep: “cop”). The auxiliary (https://universaldependencies.org/docs/u/dep/aux_.html) and case marking (https://universaldependencies.org/docs/u/dep/case.html) of the link verb will also be in the verb part of the SVO collocation. For example, in the sentence “I was in Roppongi one time with Koji.” (Ghostwritten, David Mitchell, 1999), “in” is the case marking of “was” and the verb part in the SVO collocation is “was in”.

Clausal Modifier

From https://universaldependencies.org/docs/u/dep/acl.html we read: “[The adverbial clause] acl stands for finite and non-finite clauses that modify a nominal. The acl relation contrasts with the advcl relation, which is used for adverbial clauses that modify a predicate. The head of the acl relation is the noun that is modified, and the dependent is the head of the clause that modifies the noun. A verb as a clausal modifier may not be the syntactic head of the subject as usual, so their subjects need extracting differently. In this sentence “Their hair was the same length, their lipstick the same color, their bodies curving in the same way beneath their same uniform.” (*Ghostwritten*, David Mitchell, 1999), “bodies” is the subject of “curving”, and “curving” is also in a clause that modifies “bodies”. In the output of dependency parser, “bodies” is the syntactic head of “curving”, which is its clausal modifier (dep: “acl”). Nouns that governs a verb with the dep of “acl” is recognized as its subject. When a noun governs a clausal modifier verb, that noun

will be that verb's subject in the output. Otherwise, the real subject will be omitted due to its absence in the verb's govern dictionary and will be substituted by the default subject "Someone?". However, sometimes Stanford CoreNLP can not accurately tag a clausal modifier verb as "acl". For instance, in the sentence "She wrote the seat number on my boarding pass and returned it with the rest of my papers, looking at me for the first time with grape-colored eyes that were a consolation until I could see Beauty again." (Strange Pilgrims, Gabriel Garcia Marquez, 1992), "looking" is tagged as "dep"----dependent (de Marneffe and Manning 2008). In our approach, if in a token's govern dictionary there's a verb with the dep of "dep", it will go through the processing of acl extraction as well.

Default Subject/Object

For some verbs whose syntactical head is **another verb** the subject or object may neither be in its govern dictionary nor be its syntactical head (like in the case of clausal modifiers). In these cases, the subject or object of its syntactical head will be set as its default subject/object.

One case is when a verb is a clausal modifier (dep: advcl). From https://universaldependencies.org/docs/u/dep/advcl.html, we read: "An adverbial clause modifier is a clause which modifies a verb or other predicate (adjective, etc.), as a modifier not as a core complement. This includes things such as a temporal clause, consequence, conditional clause, purpose clause, etc. The dependent must be clausal (or else it is an advmod) and the dependent is the main predicate of the clause." If a verb is an adverbial clause modifier, it modifies its syntactical head, which is a verb, and sometimes its subject / object is the subject / object of the verb that it modifies. If the subject / object is missing in its govern dictionary, the subject / object of the verb that it modifies would be set as the default subject / object.

For example, in the sentence "she would have to take refuge in my arms to escape her terror." (*Strange Pilgrims*, Gabriel Garcia Marquez, 1992), "escape" (dep: "advcl:to") modifies "take", and the subject of "take" is "she". In this sentence, "she" is also the subject of "escape", but neither is the syntactic head of the other. In this case, the subject of the verb which is the syntactic head of that adverbial clause modifier will be its default subject (unless the adverbial clause modifier is the syntactic head of its own subject). If the POS tag (https://stanfordnlp.github.io/CoreNLP/pos.html) of that verb is "VBN", that default subject will then become its object. In this case, if "she" is not set as its adverbial clause modifier's default subject, it will be omitted and the subject of "take" in the extraction output would be "Someone?".

If a verb is an open clausal complement (dep: "xcomp"), which is a predicative or clausal complement without its own subject, it can be extracted with default subject/object in the same way above. In our approach, if in a token's govern dictionary there's a verb with the dep of "dep", it will go through the processing of clausal complement / clausal modifier extraction as well.

A similar situation when a default subject/object within a sentence is set for verbs which are not the syntactic head of their subjects or objects is the conjunct (dep: "conj:xxx") (de Marneffe and Manning 2008). In this sentence "He never writes or texts to me.", "writes" is the syntactic head

of its conjunct “texts” (the dep of “texts” is “conj:or”). The subject of “texts” is “He”, and its object is “me”, but the govern dictionary of “texts” is completely empty. If no default subject and object were set, “texts” would not be in the final output due to the absence of both. To maximize the accuracy, a default object within the sentence would not be the object of verbs that are behind it in the sentence, since in English the object usually follows the verb (Dryer, 2013). Without this rule, a verb may be given an object falsely. For example, in the sentence “He crossed the bridge and then ran away.”, “ran” is the conjunct of its syntactic head “crossed”, but “bridge” is not the object of “ran”. If “bridge” is still the default object for the verb “ran” that is behind it in the sentence, it will cause an incorrect SVO collocation in the final output.

Verb Semantics

Verbs can change their meaning depending upon their wider colocations or combinations of words, namely, prepositions they are linked to (e.g., “sit up” and “sit down”) and combinations of verb+noun+preposition (e.g., “take care of” which has a different meaning from simply “take”).

Verb and Preposition

In a verb-preposition-object collocation (for example: “he puts on the cover”), the object’s dep is “obl:preposition” (in “he puts on the cover,” the dep of “cover” would be “obl:on”). When multiple tokens governed by the verb have the dep of “obl:preposition”, we use the txt file (verb_prep_json.txt) in the “OpenIE” folder in the lib path, which is based on common verb-preposition collocations (https://7esl.com/verb-preposition-combinations/; https://eslgold.com/grammar/verb_preposition_collocations/), to extract the correct object: if the verb’s lemma is in the keys of the json and the preposition precedes one “obl” token is in that key’s value----a list of prepositions, that token would be selected as the object.

Light Verb Constructions (LVCs)

The NLP Suite SVO extraction algorithm accounts for LVCs, thus providing more robust SVOs.

Light verb constructions (LVCs) are verb and noun combinations that, together, provide a new and different meaning from its constitutive parts. A simple test for LVDs is to replace the complex verb (+preposition) +noun with a single verb: “She took into account” “she accounted;” “I had a dream” “I dreamed;” “He took advantage of John” “He exploited John”). The most typical light verbs that result in LVCs are: make, take, have, do, give, get (these are the verbs Tu and Roth, for instance, focus on; Tu and Roth 2011). By postprocessing LVCs, our algorithm increases the precision of extracted SVOs (e.g., S: he V: took O: advantage instead of S: he V: took advantage of O: John). For some examples in English, see https://en.wikipedia.org/wiki/Light_verb#English. For a more complete list of LVCs, see https://en.wiktionary.org/wiki/Category:English_light_verb_constructions.

Computer scientists involved in NLP are increasingly looking at ways of dealing with LVCs (Nagy et al. 2020; Calzolari et al. 2002; Sag et al. 2002).

Multi-Word Expressions (MWEs)

LVCs are part of more general forms of multiword expressions (MWEs) where a combination of words provides new meaning (e.g., compound nouns such as “kitchen knife,” “operating room”).

Collocations

LVCs and MWEs are also generically referred to as *collocations*, i.e., noticeable arrangements or conjoining of words that have specific meaning (e.g., "to save time" and "make the bed" are common collocations).

Light verb constructions (LVCs) are verb and noun combinations in which the verb has lost its meaning to some degree and the noun is used in one of its original senses, typically denoting an event or an action (Nagy et al. 2020). Some LVCs, such as “take care of”, will lose its original meaning in the final output after the extraction. For instance, in the sentence “He will take care of my cat.”, “take” is the syntactic head of “care” and “cat”. In the normal extraction process, since the dep of “care” is “obj” and that of “cat” is “obl:of”, “care” will be extracted as the object. Therefore, although the verb should be “take care of” and object should be “cat”, in the output form the verb is actually “take” and the object is “care”. Another LVC that has a similar pattern in the dependencies is “take advantage of”.

To extract similar LVCs as a whole verb, A json was constructed (the txt file “verb_obj_obl_json.txt” in the “OpenIE” folder in the lib path). That json’s keys are verbs (for example: “take”) and each value is a list of dictionaries with two values: the first key is “obj”, and the second key is “obl” (for example: {“obl”: “care”, “obl”: “of”}). If one verb’s lemma is in that json’s keys, its object matches the first value in a dictionary in that key’s value, that verb governs a token whose dep is “obl:preposition”, and preposition matches that dictionary’s second value, then that verb, it’s object, and the preposition in the dep “obl:preposition” will be extract as one verb, while the token whose dep is “obl:preposition” will be its object. That txt file of the json can be manually edited when there’s need to add another LVC that plays the role of a single verb and has a similar dependency pattern: verb + object + preposition, and the real object of this LVC is governed by the verb with the dep “obl:preposition”.

Future Development for LVCs and MWEs

Some LVCs have different dependency patterns, such as “take charge of”. In this sentence “The candidate will take charge of every stage.”, the verb “take” is not the syntactic head of the actual object “stage” (dep: “nmod:of”; its syntactic head is “charge”). Therefore, its extraction cannot depend on “verb_obj_obl_json.txt.”

LVC is a subset of Multi-Word Expressions (MWEs), which refer to various types of linguistic units or expressions, including idioms, noun compounds, named entities, complex verb phrases and any other habitual collocations (Tu and Roth, 2011). Some MWEs also needs extracting as a whole verb, such as “be in charge of” and “be responsible for”. For MWEs with irregular dependency patterns, one potential method to extract them is to collect enormous English sentences and ask volunteers to label the index of tokens that are subjects, verbs, or objects. If there are MWEs that are labeled as one verb, a graph of these tokens’ dependency relationships

would be generated by breadth-first search, and specific json can be generated from these graphs.

References

TIPS_NLP_Stanford CoreNLP parser.pdf
TIPS_NLP_Stanford CoNLL table.pdf

Klein**,** Dan and Christopher D. Manning. 2003. "Accurate Unlexicalized Parsing." *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pp. 423-430.

de Marneffe, Marie-Catherine and Christopher D. Manning. 2008. Stanford typed dependencies manual. Revised for the Stanford Parser v. 3.7.0 in September 2016. https://downloads.cs.stanford.edu/nlp/software/dependencies_manual.pdf

Matthew S. Dryer. 2013. Order of Subject, Object and Verb. In: Dryer, Matthew S. & Haspelmath, Martin (eds.) The World Atlas of Language Structures Online. Leipzig: Max Planck Institute for Evolutionary Anthropology. http://wals.info/chapter/81

Nagy, István T., Anita Rácz, and Veronika Vincze. 2020. "Detecting Light Verb Constructions across Languages." *Language Engineering*, Vol. 26, pp. 319–348.

Tu, Yuancheng and Dan Roth. 2011. "Learning English Light Verb Constructions: Contextual or Statistical" *Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World (MWE 2011)*, pp. 31–39, Portland, Oregon, USA, 23 June 2011.