Variable Annotation in Premises of Argumentative Schemes

The task of variable annotation in premises consists in annotating the variables that should compose a set of premises for a given argument. To this goal, the annotator is given an instance of an argument (1), and a set of premises from a given argumentative schema (2) with a set of empty variables. Then, for each variable, the annotator will be asked to replace the variable with the relevant span of text from the argument (1) . See for example (3) and (4).

Example 1:

(1) *"Your doctor, Dr. Smith, has told you to put sunscreen on."*
(2) <*subjecta*> is in a position to know whether it is the case that <*eventA*>.
<*subjecta*> asserts that it is the case that <*eventA*>.
Conclusion: <eventA> is the case."
(3) <*subjectA*>: ***Dr. Smith***
(4) <*eventA*>: ***you should put sunscreen on***

Guidelines for the annotation:

1. The name of the variables indicates what kind of information should be in that position:
    a. *subject*: implies that the variable is the subject of an action. It is often a person, but it can also be a group or other nouns. Sometimes, the subject might be the speaker. In this case, you can just write "speaker".
    b. *event*: it is often an action to be taken, but it can be other kinds of events.
    c. *argument*: it has to be some kind of statement/opinion.
    d. *value*: it is usually a belief.
    e. *large_majority*: it should refer to a group of people.
2. There are 2 variables which can only be a closed set of options:
    a. <*direction*>: this can only be "positive" or "negative".
    b. <*neg*>: it's either "not" or an empty variable. This is the only variable that can be left empty (Example 2)
3. The variables have to be filled with the information given in the argument. However, it does not have to be a copy-paste of the content in the argument.
    a. Changes that can be done to make the premises look as grammatical as possible. There is no closed set of such allowed changes. However, some common ones are:
        i. Change the verbs to gerund form (Example 4 below)
        ii. Put the verbs in a future form (Example 5 below)
        iii. Add verbs that were not there (Example 1)
        iv. Negate some event (Example 6 below)
        v. Put the names that are in full capitals (e.g. CLINTON) to its original form (e.g. Clinton).

        vi. etc.
    b. Some variables might be implicit (Example 3)
4. Unless they start with a proper noun, annotations should start in lower-case.
5. USE OF CONTEXT: It might be the case that the argument on its own lacks enough context for the variables in the premises to be filled. In this case, the annotator can look at the CONTEXT. If the CONTEXT is useful the annotator should indicate it has used the context by marking as “yes” the variable USED_CONTEXT?. If the information is still not enough, then the annotator should mark the example as “lack of context” in the INVALID column.
6. It might be the case that the argument does not fit with the set of premises assigned to it. In these cases, the annotator should mark the example as “does not fit” in the INVALID column and leave the variables empty. The label “does not fit” should be given to all the variables in an argument.
7. However, before marking it as “does not fit”, the annotator should make an effort to make it fit. If the annotator is not sure if they overinterpreted the example, the label “difficult example” can be given in the column INVALID. This label is just for cases in which the annotator ended up filling the variables but is not sure about the result.
8. Don’t worry if the final premise has ungrammatical errors such as:
    a. Concordance: The number of the nouns does not match the one of the verbs. (e.g. "The judges is expert in domain law.")
    b. Double negation: There is a double negation (e.g. “Not going to the gym should not occur.” and Example 6)
9. In the column SCHEMA there is a set of labels that indicates the pattern that the premises are following. It is not necessary to understand the labels. However, there are labels with special rules:
    a. The SCHEMA “Example” should show arguments in which an example of an event is being used to generalize to similar events. However, the annotator might often find arguments in which there is no such structure. It has been identified that many arguments labeled as “Example” are either an exposure of evidences (e.g. “We have spent a lot of money, I have only 2 euros left.”); or cases in which there is no argument. These cases should respectively be labeled in the column EXAMPLE as “evidence giving, not example”, and “default inference”.
    b. Also in SCHEMA “Example” you’ll see there are two different possible sets of premises: (1) ExampleToGeneralisation: “Trump is a politician and a liar. Therefore, all politicians are liars.” or (2) GeneralisationToExample: “All politicians are liars. Therefore, Clinton is a liar.” → which of these subtypes is the case should also be marked in the column EXAMPLE.
        i. ExampleToGeneralisation: the goal of the argument is to say that, since an example (Trump) with feature 1 (politician) had feature 2 (liar), other examples with feature 1 will also have feature 2.
        ii. GeneralisaToExample: the goal of the argument is to say that, since most examples that have feature 1 (politicians) also have feature 2 (liars), then this example (Clinton) that has feature 1, will probably have 2.
10. When possible, variables should only use pronouns when the subject has been mentioned in that individual variable already. For instance, in the sentence “The more ISIS grows, the more difficult it becomes to defeat them” is a Consequence

Argumentation Scheme. The first event is “ISIS grows more”, and the second one is “it gets more difficult to defeat them”. However, we will annotate “ISIS gets more difficult to defeat” or “it gets more difficult to defeat ISIS”. This rule won’t apply to feature variables (<featG> and <featF>).

11. Once all the instances have been labeled, the annotator should do a second round to review the cases that have been marked in the INVALID column. This second round has the goal of reducing the number of instances with no variables.

Additional examples:

Example 2:

SCHEMA: “Consequences”
You should put sunscreen on, otherwise you’ll get burnt.
If <eventA>, then <eventB> may plausibly occur.
<eventB> has a <direction> consequence.
Conclusion: <eventA> should <neg> occur.
<eventA>: ***you don’t put sunscreen on***
<eventB>: ***you’ll get burnt***
<direction>: ***negative***
<neg>: ***‘’***

Example 3:

Ikea has moved its production to Thailand, and others might too.
SCHEMA: “ExampleToGeneralisation”
In this particular case, <subjecta> has property <featF> and also property <featG>.
Therefore, generally, if another subject has property <featF>, then it also has property <featG>."
<subjecta>: ***Ikea***
<featF>: ***be a big company***
<direction>: ***move its production to Thailand***

Example 4:

SCHEMA: “Practical Reasoning”
To lower the price of rents we have to put a fair regulation in place.
There is the goal of <goalG>.
Carrying out the action of <eventA> is a means to realize <goalG>.
Therefore, <eventA> ought to occur.
<goalG>: ***lowering the price of rents***
<eventA>: ***putting a fair regulation in place***

Example 5:

SCHEMA: “Consequences”
Our employment policies are a disaster, we are putting people out of work.
If <eventA>, then <eventB> may plausibly occur.
    <eventB> has a <consequence> consequence.
    <eventA> should <neg> not occur.
<eventA>: ***our employment policies are a disaster***
<eventB>: ***people will keep being put out of work***
<consequence>: ***negative***
<neg>: ***‘’***

Example 6:

SCHEMA: “Danger Appeal”
We need to be very brave to not fall into their game.
If <eventA>, then <eventB>.
    <eventB> is a danger.
    Therefore, you should not <eventA>.
<eventA>: ***not brave***
<eventB>: ***fall into their game***