Benedikt Szmrecsanyi and Lars Hinrichs

MANUAL FOR CODING RELATIVIZERS IN THE BROWN CORPORA

Intro

Dependent variable and context
We are marking up the Brown corpora for just three relativizers: WHICH, THAT, and ZERO, and only in contexts where they are 'interchangeable': in restrictive relative clauses and in non-pied piping contexts.

Marking up tokens
You, the coder, are charged with manually inserting a tag before the beginning of the antecedent NP and after the end of the relative clause end. In addition, for WHICH and THAT you need to code for the grammatical function of the relativizer (subj or obj) - see below on how to code for this.

Data and folder structure
As a general procedure, you should first download and unzip the data. After that, create three nested folders and name them 'ZEROcoded', inside it 'THATcoded', and inside it 'WHICHcoded'. You will open files from the downloaded data, mark them up for ZERO, and save the coded file within ZEROcoded/. Later you will open the files in ZEROcoded/, mark them up for THAT, and save those in ZEROcoded/THATcoded/. And so on.

How to insert tags
For all tags you insert, make sure that they are surrounded by whitespaces, and that you exclude punctuation from the strings you code. For example, when a comma or period follows a relative clause, it should be excluded from the unit you're coding, as well as its tag (see examples below).

Coding ZERO

The zero relative cases that were automatically retrieved have already been hand-sorted, so they're all keepers in principle. If there should be a false case, simply remove the <w ZR> tag.

Before beginning of antecedent, insert: <$Z>
After the last word of the relative clause, insert: <%Z>

Example:
<$Z> <DB>all **<w ZR>** <PPY>you <VM>need <VVDI>do **<%Z>** <VVBZ>is <TO>to <VVI>remember <AT>the <MC>four <NN2>names <->- <CC>and <AT>the <NN1>order <II>in <DDL>which <PPHS2>they <VV0>come <.>.
[Note: the WHICH-token contained in this example would be excluded from analysis.]

Coding THAT

Search for the string <WPR>that.
If you find that the token was inaccurately tagged as a relative pronoun, exclude it from analysis by changing the tag to <WPRX>.

Change the <WPR> tag to reflect the grammatical status of the pronoun:

If THAT acts as object of the relative clause, change tag to <WPRO>
If THAT acts as subject of the relative clause, change tag to <WPRS>

Before beginning of antecedent, insert: <$T>
After the last word of the relative clause, insert: <%T>

Example:
<II>at <RL>home <PPHS1>he <VVD>defined <PPH1>it <II>as **<$T>** <AT>the <NN1>sort <IO>of <NN1>partnership **<WPRS>**that <VVZ>exists <II>between <AT1>a <NN1>rider <CC>and <APPGE>his <NN1>horse **<%T>** <.>.

Coding WHICH

Search for the string <DDL>which.
If the token is preceded by a comma, semicolon, dash, or by a preposition, then exclude it from the analysis by changing the tag to <DDLX>.
If the tag should have been assigned in error, do the same to exclude the token from the analysis.

If WHICH acts as object of the relative clause, change tag to <DDLO>
If WHICH acts as subject of the relative clause, change tag to <DDLS>

Before beginning of antecedent, insert: <$W>
After the last word of the relative clause, insert: <%W>

Example:
<PPH1>It<VVBZ>'s <JJ>true <CST>that <NN2>roses <hi><VAD0>do</hi> <VVI>fall <NN1>victim <II>to <NN1>disease<c YCOM>,
<RR>especially <II>to **<$W>** <NN1>fungus <NN2>attacks **<DDLS>**which <VV0>damage <APPGE>their <NN2>leaves **<%W>** <,>, <CCB>but <PPH1>it<VVBZ>'s <RR>also <JJ>true <CST>that <PPY>you <hi><VAD0>do</hi><XX>n't <VVHI>have <TO>to <VVI>spray <PPHO2>them<.>.

Coding other WH-pronouns

WHICH, THAT, and ZERO is all we're coding, and only in the contexts stated above. We are not doing WHY, WHOSE, WHOM, WHERE, or anything else.