Hands-on session #1 – CorpusStudio: starting your search and developing your query

Instead of starting with CorpusStudio straight away, we will first have a brief look at Cesax.

Lesson content:

1)	Cesax: preparing the program	(10:00 – 10:10)
2)	Cesax: look at some data	(10:10 – 10:30)
3)	Cesax: preparing for CorpusStudio	(10:30 – 10:40)
4)	CorpusStudio: setup	(10:40 – 10:50)
5)	CorpusStudio: query execution	(10:50 – 11:00)
6)	CorpusStudio: your own query!!	(11:00 – 12:00^{+})
7)	Sharing results *(time permitting)*	

1 Cesax: preparation

1) Adjusting Cesax' settings to your environment
   a) Starting the program for the first time
   b) Tools/Settings
      i) Working directory: location of the *psdx* files I want to look at
      ii) Period definition file: choose
      "My documents" / ru / CorpusStudio / eng_sla / Periods LONGDALE workshop.xml
      iii) Chain dictionary: accept default location (this is used for coreference resolution)
      iv) CorpusStudio directory: (leave for now)
      v) Press "Apply" and exit the Settings window

      vi)

2 Cesax: look at some data

1) Loading a *psdx* file
   a) Load a file using File/Open or Ctrl+O
   b) Move to your "Longdale_2014/Longdale/Longdale Hannover" cohort 1, W004
   c) Open file LUH001-W004.psdx

2) Looking at the syntax of a simple sentence
   a) Put your cursor in line #14 (or do Ctrl+G, and type "14")
   b) You are with the sentence: *It became a part of my everyday life*
   c) Now go to tab page "Tree" → you get a tree representation of the same sentence
   d) Look at tab page "Syntax" → you see the bracketed labelling format of the sentence
   e) Look at the print of how the sentence is coded in the *psdx* file (ref), and note:
      i) `forest`: every sentence is contained in <`forest`> … </`forest`> tags
         (1) Attributes: `forestId`, `File`, `Textid`, `Location`
      ii) `eTree`: each phrase (such as S, NP, VP) as well as word category (such as PRP, VBD) is contained in <`eTree`> … </`eTree`> tags
         (1) Attributes: `Id`, `Label`
      iii) `eLeaf`: each endnode (leaf) holds the actual text in <`eLeaf`> … </`eLeaf`> tags
         (1) Attributes: `Type` (may be *Vern* or *Punct*), `Text`
   f) Syntax of the simple clause:
      i) S → NP_{SBJ} + VP ("sbj" and "obj" are not indicated on the NPs)
      ii) VP → VBD + NP_{OBJ} (VBD = past-tense verb)

3) Relative clauses
   a) Goto line #6
   b) Look at the subject NP in tree view
   c) Shrink the whole VP (press on the minus-sign under "VP")
   d) RC consists of:
      i) NP_{main} → NP_{head} + SBAR
      ii) SBAR → $WHNP_{which}$ + S — *the linguistics, which I just started learning …*
   e) N.B: alternative SBAR possibilities
      i) SBAR → $WHNP_{that}$ + S — *something **that** might persuade me to get a room*
      ii) SBAR → $WHNP_{who}$ + S — *people **who** interest me*
      iii) SBAR → $WHPP_{in\text{-}which}$ + S — *the language **in which** I can express my feelings*
      iv) SBAR → $WHADVP_{why}$ + S — *a reason **why** I'm good at English*
      v) SBAR → $WHADVP_{where}$ + S — *a website **where** student rooms are put up*
      vi) SBAR → S — *the English accent ___ I find very interesting*

3 Cesax: preparing for CorpusStudio

1) Preparing a search for relative clauses
   a) Be in the tree-view of line #6
   b) Follow the instructions, and select (in this order) the following constituents:
      NP (top one) – (no name needs to be given; this becomes `$search`)
      NP (containing 'the linguistics') – name it as '**preNP'**
      SBAR – name it as '**rc'**
      WHNP – name it as '**compl'**
   c) Delete the condition [3 ___ Order search descendant ___ phrase]
   d) Press "Export"

4 CorpusStudio: setup

1) Adjusting CorpusStudio's settings to your environment
   a) Launch the program (N.B: wait until the status has *"Settings have been loaded from … "*)
   b) Tools/Settings
      i) Working directory: location of the *psdx* files you want to look at
      ii) Project editor: don't touch unless you know what you are doing!
      iii) Language editor:
         (1) This information is taken from the internet after startup
         (2) If you would like to add a language (variant), period definitions and/or Xquery definition files, and make them available for others: send them to us!
   c) Press "Apply" and exit the Settings window

2) Create a new corpus research project
   a) Create a new project (File/New or Ctrl+N)
      i) Name: `LongdaleTest` *do not use spaces in names!*
      ii) Project type: `Xquery-psdx`

3) Use the project initialisation wizard
   a) This is not obligatory! If you don't want to use it, check "Do not use the project initialisation"
   b) Tab page "Language"
      i) We will use: [`eng | English | sla`] (second language acquisition // learner's English)
   c) Tab page "Periods/genres"
      i) Use "`Periods LONGDALE workshop.xml`"
   d) Tab page "Definitions"
      i) Check "I would like to specify a main definition file"
      ii) Select "`Sanne_xq-def.xq`"
      iii) Uncheck "Add one or more definition files (with functions)
   e) Confirm project initialisation: press "Ok"
   f) Save the project: Ctrl + S *Save it in an easy to find directory*

4) Checking and changing file locations
   a) Open tab page "Files"
   b) Select "Suggest and create Query and Output directories"
   c) Change the "Input directory" to the directory containing:
      Longdale_2014/Longdale/Longdale Hannover" cohort 1, W004

5) Providing meta information for this project
   a) Go to tab page "General"
   b) Give the "Goal" (purpose) for this project succinctly
   c) Optional: comments on how you intend to reach the goal
   d) Project is locked: safest for now…

6) Double checking
   a) Go to "Period editor" → is this loaded?
   b) Go to "Definitions" → is "Sanne_xq-def" loaded??
   c) Save the project so far (Ctrl + S)

5 CorpusStudio: query execution

1) Using the query wizard
   e) Go to tab page "Query Editor"
   f) From the menu, select "Query/Wizard"
   g) Query creation wizard:
      i) Tab page "General"
         (1) Provide a name: `npRCtest`
         (2) Goal: `find all NPs with a relative clause - test`
         (3) Comments: `Created for the Longdale workshop`
         (4) Type of research project: "**`Database`**"
         (5) Check "Create a project definition file…"
      ii) Tab page "Conditions"
         (1) Select "Import conditions from Cesax"
         (2) Go to tab page "Editor"
            (a) This is where fine-tuning of the conditions is possible – we will leave it as is
      iii) Tab page "Wrap-up"
         (1) Check "This query provided sub-categorized output"
            (a) This means that we want to divide the result into different categories
         (2) Check "Use [Order] relations to subcategorize"
            (a) We want to use the [order] relation (the order of NP-SBAR/SBAR-NP) as subcategorization
         (3) Select "Suggest features"
      iv) Confirm the query wizard by pressing "Ok"
   h) Double check:
      i) Has the query been created?
      ii) Has a definitions file been added (tab page "Definitions")?

2) Placing the query in the execution pipeline
   a) Go to tab page "Constructor editor"
   b) From the menu, select "Constructor/Add"
   c) Save the project so far (Ctrl + S)

3) Execute the query
   a) Choose Tools/Execute (or F10)
   b) Wait…
   c) Correct output: 147 results
   d) Click through to the results, and the first two look like this:
      i) `The linguistics;which I just started learning a few weeks ago;`**`which`**
      ii) `teachers;who tell you what to learn and when;`**`who`**
   e) The last word is the complementizer that has been found.
      i) Which ones are there????

4) Categorizing on the complementizer
   a) Go to the Query editor
   b) Type the following line before `(: Make sure only the correct ones match :)`
      ```
      let $cat2  := concat($cat, '_', ru:NodeText($compl))
      ```
   c) Change the "return" line to:
      ```
      return ru:back($search, $db, $cat2)
      ```
   d) Save the project (Ctrl + S)
   e) Run it again (F10)
   f) Look at the resulting 'complementizer' classes
      i) This shows the Stanford parser may not always have parsed the texts right…

6 CorpusStudio: your own project

1) Choose a corpus research project
   a) Make use of the table
   b) If you want to use (3), consider doing (10)
   c) If you want to do any of (1) – (5), then think of *categories* for the output

#	Query	Find	Count	Categories	Database
1	Find passives	x	x		
2	Find *it*-extraposition	x	x		
3	Find relative clauses ←	x	x		
4	Find embedded *–ed* clauses	x	x		
5	Find embedded *–ing* clauses	x	x		
6	Find periphrastic tenses	x	x	tense type	
7	Count tense forms	x	x	tense type	
8	Count aspect use	x	x	aspect type	
9	Where are adverbials used?	x	x	Adv[P] location	
10	NP: relative clause types	x	x	RC type	x
11	NP: modification types	x	x	simple, pre, post	x
12	Structure of NPs	x	x	NP type	x
13	Structure of VPs	x	x	VP type	x

7 Sharing results

Time permitting, we will share our results and experiences.

It is important to be able to exchange your data, and CorpusStudio is well suited for that. So, if there is time, we will receive a "corpus research project" (a *.crpx* file) from one of us, and see how we can run it on our own computer to verify results.