Crouton.org

 Back to main page.

Here's a quick demo of how to use Crouton. This is a preliminary version of one of my V2 scripts called V2Sum. Click on a blue line to see commentary, and click on the comment to make it go away.

```
import "Prelude.ctm"
import "PPCME.ctm"

let
  Summary = fn [x] ->
    case x of
      {sentence : {ip : String["IP-MAT", ...],
        preVerb : ManyShortest[Any],
        verb : V,
        postVerb : ...},
       id : Any} ->
         {id, ip,
          SubjectType[sentence],
          verb,
          PreVerbSummary[{preVerb}],
          PostVerbSummary[{postVerb}]}

      {{pType : String[...], ...},
       id : Any} -> {id, pType}

      {Any, id : Any} -> {id, "confused"}

  // These should agree:

  PreVerbSummary = fn [words] ->
    {case Filter[IsSubject, words] of
       {} -> "VS"
       {Any, ...} -> "SV",
     Condense[words]}

  PostVerbSummary = fn [words] ->
    {case Filter[IsSubject, words] of
       {} -> "SV"
       {Any, ...} -> "VS",
     Condense[words]}


  SummarizeMS! = fn [fileName, sumDir] ->
    case SplitPath[fileName] of
      {dir : Any, stem : Any, ext : Any} ->
      (do
        ms <- ReadAndClean![fileName]
        (let
          msSum = Map[Summary, ms]
        in
          do
            WriteBracketed![sumDir ++ PathSep ++ stem ++ "-sum." ++ ext, msSum]
            Return![msSum]))

in
do
  SummarizeMS![First[ARGS], "."]
```

Here's how to use that script. First, here are the module, script, and data files. Download all of these and put them in a folder.

Prelude.ctm | PPCME.ctm | V2Sum.ctn | example.psd

Note: The file example.psd is the first few sentences from a PPCME manuscript. example.psd ends in ".psd" so some computers think it's a Photoshop document rather than a text file. Just save it and load it in a text editor if you want to see what's in it.

To run the script go to a command line and run

```
crouton V2Sum.ctn example.psd
```

and a moment later, you should find a file called example-sum.psd:

```
((ID CMAELR3,26.4) IP-IMP)
((ID CMAELR3,26.5) IP-IMP)
((ID CMAELR3,26.6) IP-MAT (N (N apostel)) (VBP sei+t) (SV (NP-SBJ)) (SV (, QTP E_S)))
((ID CMAELR3,26.7) IP-MAT-SPE)
((ID CMAELR3,26.8) IP-MAT (PRO (PRO hit)) (BEP is) (SV (NP-LFD , NP-SBJ-RSP)) (SV (NP-OB1 E_S)))
((ID CMAELR3,26.9) IP-MAT (N (NPR Crist)) (VBP sei+t) (SV (CONJ PP NP-SBJ)) (SV (PP , QTP E_S)))
((ID CMAELR3,26.10) IP-IMP-SPE)
((ID CMAELR3,26.11) CP-QUE)
```

You can see that several sentences were imperative (IP-IMP), direct speech (IP-MAT-SPE, IP-IMP-SPE) and questions (CP-QUE) that didn't fit the first pattern in `Summary`, so it fell through to the second pattern. Other interesting things that show up when you run this script on the whole file: The word *that* can be a pronoun, a complementizer, or a determiner, but when *that* is used as a subject, it's still coded as a determiner. There are lots of questions, imperatives, and speech.

Now, this is pretty complicated. Let me give you a quick explanation of why I prefer this to Corpus Search 1. First, CS1 allows you to express things like subject before or after verb pretty easily, but since I need to split everything up depending on whether the subject is a pronoun and what the pre-posed constituent is, etc, I found myself writing zillions of query files to split the corpus into different branches and ... well, the amount of programming got out of hand in a hurry. I wanted it to be repeatable so I could give some simple instructions if a linguist wanted to verify my results, and with all the Makefiles (yes, I tried using make) and the fact that CS1 can't do "or" (now available in CS2) ... well, it got complicated. At least as complicated as V2Sum.ctn, probably more so. And I wasn't very confident that I was doing things right. In fact, I know I was getting things wrong because of the *that* problem. So, I decided to try writing my own search program. I do a lot of work in Mathematica and Haskell and I decided to borrow some ideas from both of them (mostly Haskell) to design Crouton.

Last modified: Thu Jun 23 13:58:06 EDT 2005