ChatScript-Data-Loading

Overview

The CS engine is a collection of capabilities which requires data to exploit. On startup the system loads cs_init and cs_initmore, dictionary, LIVEDATA, and TOPIC data.

CS_INIT files

Two init files define bot variables, limits, hookups to outside, and other things. Run-command parameters are processed first, then cs_init.txt, then cs_initmore.txt. If any parameter is redefined by a later collection, that last one wins. cs_init.txt is used for parameters that are machine-dependent and vary based on where cs is being started. cs_initmore.txt is used for globally applicable parameters.

Dictionary

For each language, CS reads in DICT/... data. Initially it loads the .txt forms of data and then writes out .bin forms (dict.bin and facts.bin) for faster loading on restarts. When only 1 language is requested, the bin forms go in the folder for that language. If multipe languages are requested, the bin files go in the top level DICT folder. Japanese language has no cs-visible dictionary. Where multiple languages have the same spelled word, each gets its own entry with appropriate language bits to differentiate.

A sample english dictionary entry is this:

```
march ( meanings=12 glosses=11 NOUN_ABSTRACT VERB_PRESENT
   VERB_INFINITIVE NOUN VERB NOUN_SINGULAR COMMON4
       KINDERGARTEN posdefault:NOUN VERB_DIRECTOBJECT VERB_NOOBJECT )
    march~1nz (^progress~2) a steady advance
    parade~5v
    demonstrate~4vz march in protest
```

The word, then in parens various flags (properies, systemflags, internalbits) along with a count of how many meanings it has and how many glosses it has. There may be multiple parts of speech for this word in the flags. In the subsequent lines each meaning is given. A meaning identifies its part of speech and links (usually) to a synonym. The synonym (eg demonstrate~4) is which entry of the demonstrate word has the same meaning. n and v here refer to noun and verb postags. the z indicates that OUR definition here is considered the master definition for the synonym set (synset). Words often are more specific instances

of general words, like dog is a canine is a mammal. Those ISA links will be facts linking the master of the more specific to the more general. This allows us to mark appropriately higher words when we see a lower one. Here, for march~1 (1st definition) it is a synset head and has an uplink to progress~2. Progress~2 is the entry "onward_motion~1nz (^motion~1) the act of moving forward" which in turn uplinks to motion~1. So when we see the word march (as a noun) we will mark it and progress and motion.

A sample spanish dictionary entry is much simpler. It does not come from Wordnet (which has structure and glosses and synsets). It comes from Treetagger's simple data.

imprimibles (ADJECTIVE ADJECTIVE_NORMAL) lemma=imprimible

impuestos (VERB_PAST_PARTICIPLE NOUN VERB NOUN_SINGULAR NOUN_PLURAL) lemma= impuesto|imponer NC VLadj

For english we don't have conjugated forms in the dictionary, we compute their lemma on the fly. For spanish the dictionary lists all forms of the word and provides the name of the lemma. Words that represent multiple parts of speech can require different lemmas, so the lemma data indicates what the lemma is for each part of speech. Treetagger lists parts of speech tags differently from english (often more complex). The flags inside parens are mappings to english pos. The lemma data keeps the actual tags from treetagger. The top level of Dict has spanish tag documentation file that explains what the spanish tag means and a file spanish tags that maps the spanish tag to a list of cs tags. their english equivalent where known. Eg,

```
NC Common nouns (mesas, mesa, libro, ordenador)
NC NOUN NOUN_SINGULAR NOUN_PLURAL
```

Note that NC spanish says nothing about singular or plural. Treetagger doesn't care. We have our own code to detect plurality, gender, and pronoun referencing.

Livedata

There is no binary form for this data. It includes global data and language specific data. Global data includes the names of top-level internet domains, definitions of predefined query types, and system- controlled punctuation behavior. Language specific data includes the rules for pos-tagging (english only), spelling substitutions that come with cs (as opposed to replace: ones listed in your scripts), explicit canonicals, handling for currency, numbers, months.

TOPIC

This folder contains subfolders BUILD0 and BUILD1. :build 0 makes the content of BUILD0 and :build botname makes the content of BUILD1. These are loaded in order. The text files are read, then rewritten out as .bin files for faster loading

in the future. Files are canon, describe, dict, fact, keywords, macros, map, patternwords, private, script.

Canon

Canon are lemma (canonicals) explicitly defined in script using canon: .

```
does-1 do-1
```

The above says that the word does has a lemma of do in the first language in the language= list (english).

Describe and Map Describe and map are documentation/debug files, not relevant to running code.

Dict Dict is supplemental words with optional flags as found in concept definitions. eg. `+ authoring VERB_PRESENT_PARTICIPLE VERB ADJECTIVE_PARTICIPLE NOUN_GERUND GRADE3_4`

Fact

Fact includes global variables and their value, then facts created.

```
$awe_talks=ja-+0
#`end variables
( ~normalcreature component ( head count 1 x00100000 ) x00100020 )
( 323i_Touring instance BMW x00100001 256 )
```

In the above, the value of $awe_talks is set to a json array. Then there is a complex fact. Subject and verb are simple but object is itself a fact. Each fact has fact flags. When a fact needs to designate it is visible to only a specific bot, it may have a tailing bot id number (like 256 above).

Keywords

Keywords are the list of concepts and implied concepts (topic) along with their keywords and flags on the concept. A T in front of the ~ indicates this is a topic, not just a concept.

```
~musicalbums NODUPLICATE ( ~musicalbums_ja_jp ~musicalbums_en_us )
~creditcardlist NODUPLICATE ( 'Discover`1048576-1
    Visa`1048576-1 MasterCard`1048576-1  Visa`1-1 MasterCard`1-1 )
```

Some concepts allow repeated words and some don't. If the words are simple, they are delimited by space in front and back (see ~musicalbums). Complex words use a backtick at the end followed by the additional complexity of possibly what bot id is allowed to see the membership and possibly what language the word is from. Note above has two copies of Visa and MasterCard, for 2 different bots. And note Discover has a quote in front of it.

Macros

Macros are the functions defined in script.

```
^setcategory 0 1 0 D( ^category $_specialty $_allowpivot $_category  )
    ^if 00G( $holder.n) `
```

It has function name, kind of function (O = outputmacro), bot id, flags on the arguments, number of arguments, list in parens of argument names as well as all local variable names, followed by compiled code.

Patternwords

Patternwords is a list of words to be marked with the PATTERNWORD flag. They are words not known in dictionary which arise in patterns and for which we don't want spellcheck to find some dictionary word to replace it.

Private

private is a list of word pairs from replace: , with original and replacement words.

Script

Script are the TOPICS definitions. The first line tells how many topics, when this was compiled for what build and with what version of cs.

```
00081 May31'22-16:41:25-1654011685478 xander 12.1
```

Successive pairs of lines are topic data.

First line of pair is name, flag bits, checksum, number of top level rules, number of gambits, byte size and what file it is defined in.

Second line of pair is a text string of all the botnames allowed to see this topic, and then its compiled rules. Each rule ends in backtick. Followed by the next rule. The first 3 digits of a rule are the jump code that translates to an offset where the next rule begins. A zero offset will happen at the very end of all rules

```
TOPIC: ~services_en_us 0x3 99311429 18 0 2406 services_en_US.top
" all " 00y u: ( ) ^check_lang ( en_US ) `00B u: ( [ vine ] )
    ^fail ( TOPIC ) `000
```