6.905 Project Proposal

Abdi-Hakin Dirie & Jason Tong

Domain

Linguists develop new theories about how language works by investigating the presence of certain phenomenon in large corpora of data. For example, a linguist might study child language acquisition and try to find data that demonstrates children have the tendency to over-regularize the past tense of a verb that has an irregular form (e.g. using “goed” for past tense of “go” instead of “went”) at a particular stage of development. In order to do so effectively, linguists must be able to efficiently search through enormous corpora of data like the ones found in CHILDES (Child Language Data Exchange System). It contains annotated transcripts of recorded conversations with children. The goal of this project is to produce an extensible library of functions that allow linguists to more effectively browse the data found in CHILDES.

Decomposition

CHILDES provides its data in a uniform format. The first component of this system is therefore a module that can ingest a collection of these files. The rest of the components are modules specific to browsing various types of annotations, including but not limited to phonological, morphological, grammatical notes accompanying the conversations. The ability to add these modules to the system is what makes the system potentially very powerful. One basic functionality that would intuitively be expected to exist is key-word matches. Therefore the system should have a module with functions for searching through the corpora for some pattern of words using boolean search. Linguists should also be able to specify whether those matches occur in child production, or if they desire to include utterances made by parents and investigators. As a proof of concept for linguistic utility, we find it reasonable for this project to include a module that matches some regex pattern of IPA (International Phonetic Alphabet) symbols, which allows linguistic researchers to discover phonological phenomena. Possible extensions include matching for syntactic structure, and searching for morphological phenomenon.

Implementation

We will store the plain text source files in a specified directory. For now will store the files as plain text when we search. We are also considering ingesting and parsing each document into a special data structure that contains metadata on the document to allow for more efficient search.

Next, we must formulate how a query is structured. Our system will employ regular expression as the low level search mechanism. However, the user will need to specify what the linguistic or annotative domain (e.g. syntax or actions) they wish to explore. Furthermore, multiple regex patterns may be used in a single query, combined in a combinatorial way. The below example returns a list of all documents that have the comment “rocks chair” and phonological representations of either “toy” or “hi”.

```
(search (and (comment (re:quote “rocks chair”))
             (phon (re:alt (re:quote “tɔɪ”)
                            (re:quote “haɪ”))))))
```

Example query

This introduces the next step in the implementation: logical search. For example, a query that conjoins two regex patterns should return only the documents that match both patterns. This will involve implementing more expressive data structures beyond just regular expressions. Queries are naturally expressed as trees, thus evaluation can be handled be evaluating the leaves and doing set arithmetic (via generic operations) up the tree. Queries (and subqueries) will always return a list of documents that match.

In the situation where a query matches multiple documents, we are still left with the task of ranking them. The problem of ranking can be handled in multiple ways, depending on the ranking metric. We will initially do no ranking (i.e. return all docs that match the query in an unspecified order). A possible extension is a weighted rank based on which document matched the most regex expressions.

The conversational data is usually subdivided into utterances followed by accompanying annotations, including comments about the context or action taken by the subject while making the utterance. We might want to preserve this sequential representation so as to support more complex queries that can operate at this finer level of granularity, another possible future extension.