LING570 Distributional Learning
Problem Set 1: Locating Word Referents
Due Oct 5

(Team work is encouraged but must be acknowledged in the submitted report which must be written independently.)

Your task is to implement several word learning models, execute them on an annotated corpus of infant-directed English, and report performance results. To do so you need to read the Pursuit paper (aka Stevens et al. 2017) closely, and you may also find it useful to consult the Supporting Information of that paper available **here**. The goal of this problem set is to develop an appreciation of the mechanical nature of language learning.

The corpus was annotated by Jon Stevens from the Rollins corpus in CHILDES, and attached as a text file. In the Pursuit paper, there are two such files, one for training and the other for testing. That's the standard practice for computational modeling but we are combining them for simplicity.

Each utterance is represented by two lines. For instance:

```
hey there little boy im a little pig
CAR RING PIG

look here look at this these are little feet
CAR RING HAND FOOT
```

The first line contains the phonological words, always in lower case, that the infant heard, and the second line is the set of referents, always in upper case, that were judged to be observable by the infant and could potentially serve as meaning candidates for the words heard. So in the example above, when the word *pig* is heard, there is in fact PIG but CAR and RING are also available. It is often the case where a word is uttered, the target referent is not available at all. No other assumption is made about the nature of the phonological words or the referents. The task is simply to see how to find the mapping between words and referents as the learning model churns through the corpus that has a lot of referential ambiguity.

An additional file is included as the "gold standard" lexicon, in a separate text file. This can be regarded, by judged by the annotator, as the best any model could conceivably do given the learning corpus. It contains 17 meanings such as COW; some of these meanings are mapped to multiple words (e.g., *cow, cows, moocow, moocows*). A model is deemed to have learned COW correctly if it maps COW to any of these words as there is no plausible way to distinguish them given the data.

Your job is to implement **four** learning models: (1) Propose but Verify, (2) Cross-situational learning as formulated in the Pursuit paper, (3) Pursuit, and (4) Pursuit with sampling. Pursuit with sampling is just like Pursuit except: when the learner hears a word, it samples among the meaning candidates proportionally with respect to their probabilities, as opposed to Pursuit which deterministically goes for the candidate with highest probability. You can use the parameter values for these models in the Pursuit paper but feel free to tune your own to maximize performance.

Each learning model will be tested on the Rollins corpus file, and the lexicon it learned at the end will be compared to the gold standard lexicon, and performance results—precision, recall, and F-score—will be reported. See the Pursuit paper for the definition of these metrics. Cross-situational learning is a deterministic algorithm and only needs to be run once while the other three models are stochastic and generally produce different lexicons for each run: for these, you will need to run them many times—say, 1000—and report the mean and standard deviation for the results. **A table such as Table 1 in the Pursuit paper should be reported**.

For extra fun—but not credit, this is a graduate level class!—code up some experimental data in the same data file format and see how these models account for the results. Examples include those in the Pursuit paper: Yu & Smith (2009), Trueswell et al. (2013), Kohne et al. (2013), but also Medina et al. (2011) and many others. For these, you can just create arbitrary strings for words and references. The degree of referential ambiguity, which may vary from utterance and utterance and can be the result of many reasons, can be simply models by modifying the probabilities with which references are chosen. Note, however, you should use the same parameter values from the corpus analysis earlier, rather than tuning a new set of parameter values for the experiments. A model that needs to be adjusted for every experiment is not a model, but just a fancy summary of the experimental results. That's what *they* do.

Numerous extensions of these models are possible, especially when combined with behavioral experiments that can be conducted online. These may include manipulations of referential ambiguity: e.g., in most, if not all, word learning experiments, the target referent is always present when the subject heard the word, and that's clearly unrealistic. One can also deviate from the existing experiments where, in almost all cases, the referent meanings are "atomic", e.g., a single object, where in actual word learning situations, a label applies to many different objects (e.g., *chair* is used to describe 4-legged as well as 3-legged chairs, chairs with wheels, chairs with or without armrests, etc.). And we also learn words that are subordinate or superordinate to the basic-level objects typically used in word learning experiments.