Problem Set 2: Word Segmentation

Discovery Procedure, Fall 2023
Logan Swanson

Code Structure

generate_stimulus.py: This program generates training files for the two segmentation approaches, according to the specifications in Frank et al 2013. It produces the following:

- *lexicon.txt*: A list of 1,000 words, with syllable boundaries marked by spaces. Each word is on its own line.
- *stimulus_tp.txt*: A stimulus file for the wordseg implementation of transitional probability-based segmentation. It contains 15,000 lines, each of which is an utterance of at least two words, with no word segmentation information
- *gold.txt:* A test file for the transitional probability implementation. It contains the same 15,000 utterances, with spaces between words
- *stimulus_sub.txt:* A stimulus file for the WordSegmentation implementation of the subtractive method. It contains the same 15,000 utterances, each on its own line, with additional information about phoneme, syllable, and word boundaries to be used by the program (word boundaries are used for testing).

tp_segmentation.py: This program executes training and evaluation for the transitional probability method. It makes use of the wordseg package. Evaluation results are printed.

Results

Results from each of the two implementations are reported below. The most relevant metrics are for the “boundaries” category, since this is the category on which participants were evaluated in the Frank et al 2013 study. For this reason, the average value over three trials (each with a distinct randomly generated training language) is reported for this category. Variation between the trials was small for all categories (~0.05).

TP Method:

	Precision	Recall	F-Score
Boundaries (avg over three trials)	**0.81**	**.93**	**0.87**
Types (single trial)	0.19	0.80	0.30
Tokens (single trial)	0.51	0.58	0.54

Subtractive Method:

	Precision	Recall	F-Score
Boundaries (avg over three trials)	**0.65**	**0.98**	**0.77**
Types (single trial)	0.36	0.64	0.46
Tokens (single trial)	0.59	0.79	0.68
Lexicon (single trial)	0.55	0.47	0.50

Discussion

The two methods display different strengths. While the TP method has stronger performance over boundary placement, the subtractive method seems to do better with the actual words, both over tokens and types.

The participants in the Frank et al word segmentation study were evaluated over boundary placement. The four participants had different F-scores, between 0.6 and 0.8. In this sense, the TP algorithm may perform *too well* to be considered human-like. Perhaps more importantly, the human participants tended to be *conservative* with their boundary placement–generally placing boundaries accurately, simply placing too few of them. Both of these models appear to have done the opposite, with higher recall scores than precision indicating *overapplication* of boundaries (this generalization is corroborated by impressionistic examination of the segmentation output).

In this sense, I would argue neither model is capturing human behavior with complete accuracy.