Final Report

Gabrielle Ariana Ewall

July 1, 2015

Working memory is the memory that is stored briefly and manipulated. We use it to read sentences, do math problems, and remember what a room looks like when we look away. This type of memory is severely impaired for individuals with ADHD. Kofler et al. (2008) identify Working Memory deficit as the core cause of ADHD. Working memory deficits are correlated with poor academic performance (gathercole et al. 2008).

Despite the wide impact of working memory deficits, working memory itself remains poorly understood and current treatments for ADHD often do not improve academic outcomes of patients. The way that people remember lists (serial recall tasks), might reveal information about why we have working memory, and why it works the way it does. Specifically when asked to remember lists, people tend to remember the first few items and the last few items best. These effects known as primacy and recency respectively were first documented in 1885 by Hermann Ebbinghaus. We hope that exploring optimal solutions to the task of serial recall will illuminate the constraints and conditions under which primacy and recency might have developed as part of an optimal solution to a problem. We hypothesize that human working memory is optimized for a socially important task, natural language processing. To explore this hypothesis we examine the information distribution among words in a sentence, expecting the first and last words in a sentence to contain the most information content.

We used the Brown corpus which contains samples of English-language text from novels, essays, and journal articles published in 1961. We loop through a list of possible strategies, each of which is a pair of word positions to remember. We assess the quality of each working memory strategy as the negative entropy of a particular sentence given the two words kept in working memory. This value is averaged over every sentence. In other words, the higher the variation in sentences with two particular words in specified positions, the lower the quality of the working memory strategy.

Using the formula for entropy, we derive a tractable means of evaluating the quality of a working memory strategy. Because the set of possible english words and sentences is incredibly vast, we make certain simplifying assumptions along the way. We use the following notation:

$\mathcal{D} \triangleq$ database of sentences

$X \triangleq$ random variable that indicates the sentence

$X_i \triangleq$ random variable that indicates the word at position i

$\mathcal{S} \triangleq$ set of all English sentences of length m

$\mathcal{W} \triangleq$ set of all English words

The entropy of some dataset, X is given by

$$H[X] = -\sum_{x \epsilon \mathcal{S}} p(x) log\left(p(x)\right) \tag{1}$$

The usefulness of two positions in working memory is described by the amount of remaining entropy in sentences when the words at those positions are known, $-H[X|X_i, X_j]$

using the definition of entropy yields

$$-H[X|X_i, X_j] = \sum_{x \epsilon \mathcal{S}} \left[\sum_{w_i \epsilon \mathcal{W}, w_j \epsilon \mathcal{W}} p(x, w_i, w_j) log\left(\frac{p(x_i, x_j)}{p(x, x_i, x_j)}\right) \right] \tag{2}$$

The probability of sentence x,w_i, and w_j will only be nonzero when the words in the inner loop match w_i and w_j in x, which is specified in the outer loop. Therefore there is only one iteration of the inner summation which is nonzero, so the equation can be reduced to

$$= \sum_{x \epsilon \mathcal{S}} p(x, x_i, x_j) log\left(\frac{p(x_i, x_j)}{p(x, x_i, x_j)}\right) \tag{3}$$

when the sentence is x, x_i and x_j are guaranteed. Thus the expression can be reduced to

$$= \sum_{x \epsilon \mathcal{S}} p(x) log\left(\frac{p(x_i, x_j)}{p(x, x_i, x_j)}\right) \tag{4}$$

Because the set of english words and sentences is so huge, we approximate it using a sample of sentences from the Brown Corpus, $\mathcal{D}$.

$$\approx \frac{1}{|\mathcal{D}|} \sum_{K=1}^{|\mathcal{D}|} log \frac{p(\mathcal{D}_{k,i}, \mathcal{D}_{k,j})}{p(\mathcal{D}_k)} \tag{5}$$

We cannot calculate the actual probabilities of words, so we approximate by counting the number of times that word appears in the sample and dividing by the total number of words in the sample.

$$\approx \frac{1}{|\mathcal{D}|} \sum_{k=1}^{|\mathcal{D}|} log\left(\frac{\frac{count(\mathcal{D}_{k,i}, \mathcal{D}_{k,j}, \mathcal{D})}{|\mathcal{D}|}}{\frac{count(\mathcal{D}_k, D)}{|\mathcal{D}|}}\right) \tag{6}$$

$$= \frac{1}{|\mathcal{D}|} \sum_{k=1}^{|\mathcal{D}|} log\left(\frac{count(\mathcal{D}_{k,i}, \mathcal{D}_{k,j}, \mathcal{D})}{count(\mathcal{D}_k, D)}\right) \tag{7}$$

For sentences of given lengths, we calculate the value of given word position pairs in distinguishing the sentence. The combination of positions which results in the most reduction in uncertainty were considered the most important for identifying the word or sentence. In three and four word sentences, the combination of the first and last words results in the lowest conditional entropy. In a 5 word sentence, the combination of the second and last word result in the lowest conditional entropy (but this value is within s standard deviation of the conditional entropy of the combination of the first and last words). With longer sentences, this pattern disintegrates.

We apply a softmax to our results in order to illuminate which individual words are most important in the sentence. These results indicate that the last word has the highest information content, but that the first word has low information content, often lower than the words in the middle of the sentence. We find that low information content first words are frequently articles, pronouns and question words like, why and what. When sentences beginning with some of these common words are eliminated from the sample, the softmax reveals that the first and last words have the highest information content.

If we consider that each pronoun is a placeholder for a specific noun, then it is likely that our evaluation method vastly underestimates the information content of these words. For example, She carries the same contextual information as Courtney, but our algorithm considers she to be a much lower information content word because it is so common. It is also possible that natural language is not processed in single word blocks. Perhaps chunking sentences in other ways would reveal increased information content in the sentence. Consider the difference between information content distribution in the same sentence chunked in different ways:

['The','cat','walked','to','the','store']

['The cat', 'walked to', 'the store']

The first and last chunk are clearly less common than the intermediate chunk in the second example, whereas in the first example *cat* and *walked* are less common than *the*.

One direction for future work would be to examine whether our results change when sentences are chunked differently or when pronouns are replaced by the nouns they represent. Our results would be strengthened by demonstrating that a neural net using a reinforcement learning algorithm would learn to remember the first and last words in a sentence. This would allow us to eliminate the assumption that the sample from the Brown corpus is representative of the entire english language, because a neural net can handle a larger state space. It would also be interesting to see whether other languages follow patterns of positional information distribution predicted by the primacy and recency effects of by native speakers.