Computational Psycholinguistics Assignment 2

Student: Rong Ding

Number: s1067006

Part 1 Simple versus gated RNN language models

In this part I investigate the quality of simple and gated recurrent neural network (RNN) models in characterising language processing as well as how model performance developed over the training process. Model performance was assessed on two aspects, i.e. 1) how well a model captures the statistical structures in language that are useful in predicting the upcoming word, and 2) how well the output of an RNN model—in the case here, word surprisal—predicted human language comprehension behavior. To evaluate the statistical structures captured by each model, I computed and compared perplexity of word surprisal yielded by the two types of RNN models at each training corpus size. To evaluate the extent to which each model predicted human language performance, I regressed human self-paced reading data against surprisal, and examined how a model's ability to account for human reading data varied with type as well as the amount of training.

1. Method

1.1 Materials

Both test sentences and the self-paced reading dataset used in the current study come from Frank, Monsalve, Thompson, & Vigliocco (2013).

The sentence stimuli (N = 361) in Frank et al. (2013) were selected from three web novels of different genres (for details see Frank et al., 2013), which I considered as being more representative of narrative usage in English compared to those from a few other open-sourced sentence-processing datasets which are largely manually constructed and specific to experimental manipulations (e.g., Prasad and Linzen, 2019a, 2019b). To form the test set for the current study, sentence stimuli that did not meet the criteria specified in the assignment information file were excluded, resulting in a total of 329 sentences into the test set. The selected sentences were properly formatted and written into a txt-file for surprisal computation. Relevant codes that achieved test set formation can be found in *Test set selection.ipynb* attached to this assignment. See *test_set.txt* for all test sentences stimuli used in this part.

The self-paced reading dataset from Frank et al. (2013) was obtained by recording participants' button responses to words while they were reading sentences. 117 first-year psychology undergraduate students (92 females, 70 English native speakers) enrolled at a UK university participated in the study. In the experiment, each sentence stimulus was presented to participants in a word-by-word manner; concretely, each word was displayed on the centre of a computer monitor, and was replaced by next word as soon as participants made button press. Punctuations were presented together with the immediately preceding word. The time interval between word presentation and button press was recorded as the reading time on that particular word. Each sentence stimulus was preceded by a fixation cross presented centrally on the computer monitor.

1.2 Surprisal

For each word in the test sentences, surprisal was computed as the negative log probability of a word given a previous sequence of words, which can be represented with the following equation:

$$\text{Surprisal}(w_{t+1}) = -\log P(w_{t+1}|w_{1\ldots t})$$

where w_{t+1} indicates the current word, and $w_{1\ldots t}$ stands for previous words in a sentence.

Given that neural network weights were saved at 9 time points (i.e., after 1K, 3K, 10K, 30K, 100K, 300K, 1M, 3M sentences, and the entire corpus) for both types of models, a total of 18 models have been generated for examination. For each of the 18 models, surprisal of each word in the test items was computed separately using the code *get_surp.py*.

1.3 Model performance evaluation in characterising the statistical structure of language: Perplexity

The performance of each RNN model at the 9 different training corpus sizes in capturing the statistical relationships among words was evaluated by computing perplexity on the test items. Perplexity was quantified as:

$$\text{Perplexity} = 2^{\textit{average surprisal}}$$

where *average surprisal* refers to the mean surprisal over all test set words in this assignment. Lower perplexity indicates that a model assigns higher probabilities to the test data (Aurnhammer & Frank, 2019), and thus also suggests that the statistical structures of words in a language are better captured and used by the model to predict the next word. Computed model perplexities are plotted to examine the development of model quality throughout the training process (see *Perplexity computation & plotting.ipynb*).

Note that good capturing of statistical regularities in language does not necessarily entail higher predictive power in accounting for human language processing data. Therefore, I conducted a separate assessment of model performance in capturing human processing effort during self-paced reading, which I specify in 1.4.

1.4 Model performance evaluation in predicting self-paced reading data

1.4.1 Datafile creation

For each of the 18 models, a datafile with self-paced reading response times (RTs), surprisal and other variables was created with the code *Datafile_building.ipynb*. Concretely, in the first stage, word surprisal from all models was read from *surprisal.srn.test_set.csv* and *surprisal.lstm.test_set.csv* and then integrated into a csv-file named *surprisal_all.csv* containing columns *sent_nr* (sentence number acquired from *get_surp.py*), *word_pos* (word position in sentence), *word*, *model* (model type), *step* (training corpus size), *surprisal*, and *word_freq* (log-transformed mean word frequency) (see **Step 1**). Log-transformed mean word frequency comes from *Corpora from the Web* (COW14)[1], which is computed on the basis of the ENCOW14 web corpus (Schafer, 2015).

Consequently, the csv-file was reloaded into the code and read by model * training corpus size for the integration of the self-paced reading dataset with word surprisal and mean frequency (see **Step 2**). Note that, in addition to surprisal of the current word, surprisal of the preceding two words were also included; they are to be used in linear mixed-effects regression (LMER) models to control for the

[1] https://corporafromtheweb.org/

spillover effects in the self-paced reading data. An *NaN* was filled as the surprisal value of the preceding word(s) if a word was in the initial or second position of a sentence. The surprisal of words preceding a punctuation was also marked an *NaN* as the words had been presented together with the punctuation in the experiment, and thus neither word surprisal nor the surprisal of the punctuation would be accurate to represent the surprisal at those points. The integration of the data points with the variables was accomplished by matching sentence number and word position in sentence. For each model type by training corpus size, the integrated dataset was saved in a *pandas Dataframe* named *df_new*. The data points with an *NaN* in the surprisal of the current word or the preceding word(s) were removed from the dataset; in other words, words at the first two positions of a sentence and those followed by a punctuation were excluded from further analyses. The *df_new* was then index-reset and saved as a csv-file named *data_{model type}_{corpus size}.csv*.

This entire process resulted in 18 csv-datafiles in total, each of which included all of 117 participants and 242740 data points. See **Table 1** for the names of the columns and their descriptions in each datafile; note that only the variables marked bold were considered in subsequent data analyses. All the aforementioned preprocessing steps were conducted using *Anaconda 4.10.1*, *Python 3.8.5*, and the *Pandas* package (version 1.1.3).

Table 1 Names and description of variables (columns) in datafiles.

Column/variable name	Description
subj_nr	subject number
sent_nr	number of sentence
sent_pos	order of sentence in experimental presentation
word_pos	position of word in a sentence
word	word ID
RT	word reading time (defined as the interval between word onset and participants' button press)
surprisal	surprisal of the current word
surprisal_-1	surprisal of the preceding word
surprisal_-2	surprisal of the word that precedes the preceding word
wordfreq	log-transformed word frequency
wordLen	word length (number of letters)
correct	whether a participant responded accurately to the comprehension question that followed a sentence (coded as "c" (correct), "e" (wrong) or "-" (no comprehension question))
answer_time	the time a participant spent in answering a comprehension question
model	the type of the model which produced the surprisal values (coded as "SRN" or "LSTM")
step	the amount of training received by the model which produced the surprisal values (coded as *1000*, *3000*, *10000*, *30000*, *100000*, *300000*, *1000000*, *3000000*, or *8773568*)

1.4.2 Data analyses

For analysing and comparing the power of simple neural networks (i.e., SRNs) and gated networks (i.e., LSTMs) in characterising self-reading data and its development over training, two steps were

followed: first, LMER modelling (Baayen et al., 2008) was performed to regress self-paced RTs against surprisal estimates produced by each of the 18 models (i.e., 2 model type * 9 training corpus size); second, goodness-of-fit of surprisal yielded by each model was computed and compared.

In the first stage, a baseline LMER model was first fitted to the self-paced reading dataset. The goal of fitting the baseline model was to factor out as many important variables that could affect reading times as possible, and thus to leave the effect accounted for by surprisal as isolated as possible. Variables including log-transformed word frequency and word length were put in the baseline model as fixed effects to control for variance in reading times that could have been brought about by participants' familiarity with particular words or the extra cognitive effort engaged to process longer words. Sentence position, sentence number, word position and word(ID) were included as random intercepts to account for variance which had been introduced by particular word stimuli or experimental set-ups. Subsequently, regression models were fitted where target variables including surprisal of the current word as well as those of the previous two words were added into the baseline model. The model structure can thus be represented with the following equation:

$$RT \sim surprisal + surprisal\_-1 + surprisal\_-2 + word\_freq + wordLen + (1 \mid subj\_nr) + (1 \mid word\_pos) + (1 \mid sent\_nr) + (1 \mid sent\_pos) + (1 \mid word)$$

In the second stage, goodness-of-fit of surprisal produced by each RNN model was defined as the log-likelihood ratio (a.k.a. decrease in regression model deviance) between the baseline and the corresponding regression model. The measure was quantified as the chi-squared statistics with 3 degrees of freedom (resulting from the addition of three target variables, *surprisal*, *surprisal_-1* and *surprisal_-2*). Goodness-of-fit estimates were then extracted and plotted against model type and training corpus size to examine how a model's ability to account for human reading data varied with type as well as the amount of training.

All the LMER models in this assignment were implemented under RStudio (version 1.4.1103), using the package *lme4* (Bates et al., 2014; version 1.1.27). Log-likelihood ratio tests were conducted using the package *lmtest* (version 0.9.38).

2 Results

2.1 Perplexity

Figure 1 below reports the perplexities of the simple and gated RNN models at 9 different training corpus sizes. Both SRN and LSTM models display an overall decrease in perplexity over training, suggesting that the statistical structures in language become better-captured by both models as the amount of training accumulates. Note that the SRN model has lower perplexity than the LSTM does when the training corpus size is small (i.e., 1k), but the difference shrinks drastically as the LSTM model learns a few to ten thousand more sentences. The two models perform equally well in the middle of training (e.g., 100k-1M); when the training process completes the LSTM model seems to marginally outperforms the SRN model. These results indicate that the simple RNNs perform better than in capturing the statistical relationships in language when the training corpus is relatively smaller in size (say, from 1k-~30k sentences), but such an advantage fast shrinks and is even reversed as the amount of training increases.

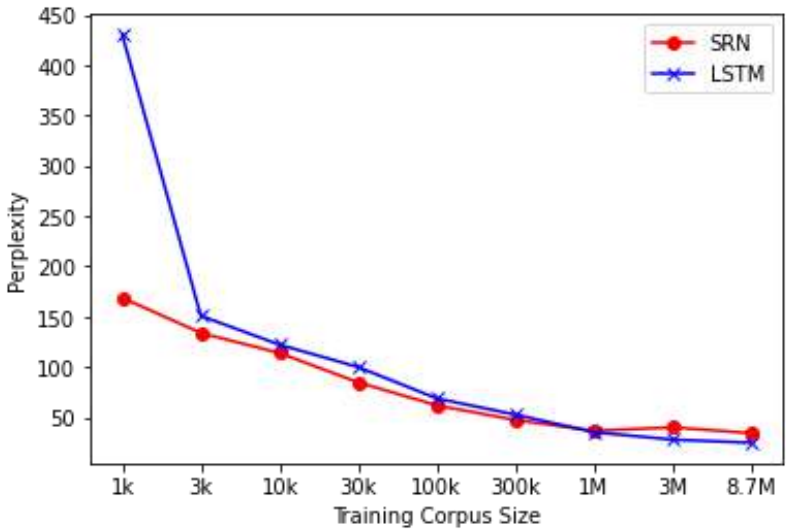

Figure 1: Perplexities of two types of RNN models, across the training process

2.2 Goodness-of-fit

The goodness-of-fit measure yields a similar yet slightly different pattern compared to perplexity (see **Figure 2**). On the one hand, the development trajectory of goodness-of-fit over training is similar to that of perplexity in the sense that both the SRN and LSTM model have their performance (fitness in this case) improved as the amount of training increases, and that the LSTM outperforms the SRN at the end of training.

On the other hand, there are two main differences between the development of goodness-of-fit and perplexity over training. First, the goodness-of-fit of the LSTM and SRN both reach a plateau at the training sizes of 3k-30k, but again drastically increase when the training size increases from 30k to 1 million sentences; in contrast, perplexity of both models increases slowly yet steadily throughout training. Second, the difference in perplexity between the two models does not positively correlate with the difference in goodness-of-fit. For instance, at the point of 3k and 10k training sizes the LSTM and SRN perform equally in predicting human reading times (i.e., equal numerically in goodness-of-fit), yet the two networks differ (though marginally in size) in terms of the ability to capture the statistical structure in language (i.e., perplexity). Conversely, the size of differences in perplexity between the two networks remains marginal when training size varies from 30k-300k, while their difference in goodness-of-fit at the training points reach the largest.

In conclusion, therefore, the goodness-of-fit of both RNN networks improves as they receive a larger amount of training, and the LSTM outperforms the SRN in characterising human self-paced reading data at the end of training. However, goodness-of-fit does not display a positive correlation with perplexity, which suggests that good capturing of statistical regularities in language does not necessarily equal higher predictive power in accounting for human language processing.

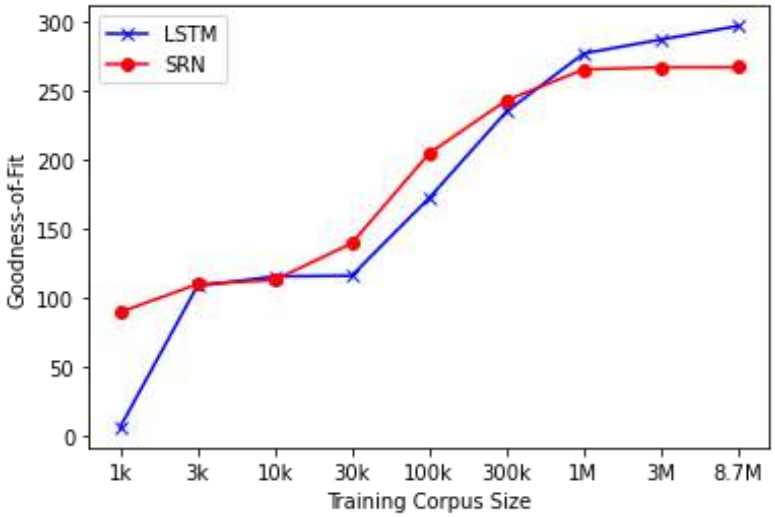

Figure 2: Goodness-of-fit's of two types of RNN models, across the training process

Part 2 Simple versus gated RNN models in accounting for the garden-path effect

In this part I examine the ability of the LSTM and SRN models to account for the garden-path phenomenon in sentence comprehension. First, I fit an LMER model to investigate the performance of the LSTM model in predicting the classic garden-path effect (that is, extra difficulties in reading are present when one alters his interpretation of a noun phrase as the second object of a verb into the subject of a following verb) as well as the model's sensitivity to the modulation of semantic variations, i.e. the goodness of thematic fit between the noun phrase and the first verb. Second, I compare the quality of the simple and gated RNN models in characterising the garden path effect, again by employing LMER modelling.

1 Method

1.1 Materials

The test sentence items used in this part come from the four files in *items.zip*. One sentence was removed from each of the bad-fit conditions because its component word(s) was absent in the vocabulary acquired by the RNN models (this was accomplished using *Sentence selection_Part 2.ipynb*). Four good-fit sentences and one bad-fit sentence where an adverb precedes the to-be-disambiguated noun phrase were additionally discarded as it would be difficult to determine the region of "critical verbs" in these cases given that preceding adverbs could have biased the disambiguation process (e.g., a facilitative effect on *never* in the sentence *Rudy disowns his husband and his family never speaks of him again*) prior to the presence of the critical verb. In the end, a total of 56 sentences were left in each good-fit sentence set and 58 sentences in each bad-fit sentence set.

1.2 Datafile creation

The creation of the datafile was achieved using the code *Datafile_building_Part2.ipynb*. In the first step, word surprisal produced by both fully trained SRN and LSTM models (computed by *get_surp.py*) was merged with word information (i.e., sentence number and word position in sentence) and conditions (ambiguity and goodness of thematic fit) into a *pandas* Dataframe (see **Step 1**). Critical verbs were picked from the Dataframe first in an automatic manner according to their relative position to the word *and*. Here we define critical verbs as the verbs immediately following a noun phrase which is initially preferred as the second object of a preceding verb in reading. The selected words were then saved into a csv-file and underwent a manual check, during which the mistakenly included words (e.g., the end of a long preceding noun phrase) were identified and replaced by the corresponding critical verbs and their relevant information.

In **Step 2** sentences/critical verbs included in subsequent analyses were first selected out according to the rules specified in the previous section. Then, surprisal estimates of the SRN and LSTM models which had originally been in two separate columns were melted into two levels of a variable named *model*. Eventually, a Dataframe with columns named *word*, *word_pos*, *ambiguity*, *fit*, *model*, and *surprisal* was saved as a csv-file for data analyses. See **Table 2** below for the description of each column.

Table 2: Names and description of variables (columns) in the csv-datafile

Column/variable name	Description
word	identity of a critical verb
word_pos	position of a critical verb in the sentence
ambiguity	the condition of ambiguity where a critical verb originates (coded as “unamb” or “amb”)
fit	the condition of thematic fit where a critical verb originates (coded as “good” or “bad”)
model	the type of model that produces the value in the *surprisal* column
surprisal	the value of surprisal

1.3 Data analyses

To investigate the performance of the LSTM model in predicting the garden-path effect, a LMER model was fitted to the surprisal estimates of critical verbs yielded by the fully trained LSTM model. Ambiguity, goodness of thematic fit and their interactions were included as the predictor variables of interest. Identity and position of critical word were put into the random structure as intercepts to account for the variance in surprisal estimates which could have been introduced by particular word stimuli or positions in a sentence. The model structure can thus be represented with the following equation:

$$surprisal \sim ambiguity * fit + (1 \mid word\_pos) + (1 \mid word)$$

Tukey’s *Post-hoc* pairwise multiple comparisons were conducted to examine the interactions between the two predictor variables of interest.

Next, to compare the quality of the SRN and LSTM models in characterising the garden path effect, LMER modelling was again performed. This time the variable *model* and its interactions with the other target variables were added into the model structure mentioned above, resulting in the equation:

$$surprisal \sim ambiguity * fit * model + (1 \mid word\_pos) + (1 \mid word)$$

To help interpret the significance of interaction effects in the model, I separately fitted a mixed model to the surprisal estimates of the SRN model. *Post-hoc* pairwise multiple comparisons were also performed to examine the interaction effects in detail. Plotting of effects of target variables was based on the separately fitted mixed models for the SRN and LSTM networks.

All the LMER and follow-up comparison analyses were performed under RStudio (version 1.4.1103), using the package *lme4* (Bates et al., 2014; version 1.1.27). Multiple comparisons were carried out using the R package *emmeans* (version 1.6.1). Codes can be found in *surp_amb_fit_LMER.R*.

2 Results

2.1 Predictions of the LSTM model

The left panel in **Figure 3** reports the predictions the LSTM model makes regarding the effects of ambiguity and thematic fit on word surprisal. The fitted LMER model shows a positive main effect of both ambiguity (β=-.02532, t=-2.185, p<.05; here *amb* is taken as zero reference) and thematic fit (β=.9728, t=2.737, p<.01; here *bad* is taken as zero reference), indicating that the lack of disambiguating commas as well as noun phrases of good thematic fit with previous verbs indeed result in less predictability of critical verbs. Crucially, LMER modelling also yields a significant

interaction between ambiguity and thematic fit (β=-.6393, t=-3.867, p<.001); *post-hoc* pairwise comparisons suggest that in ambiguous sentences critical verbs have significantly higher surprisal when preceded by a good-fit than a bad-fit noun phrase (t = -2.700, p < .05), whereas in unambiguous sentences no significant difference is observed between good- and bad-fit conditions (t = -0.930, p = 0.7889). Further details of the LMER model results can be found in Appendix (see **Appendix Table 1**).

Altogether, these results demonstrate that the LSTM model show sensitivity to the modulation of both syntactic ambiguity and semantic variations (i.e. thematic fitness) and their interactions. To be specific, when processing ambiguous sentences, the LSTM model predicts surprisal as being significantly higher in critical verbs whose preceding noun phrase fits the previous semantic context badly than in those with a badly fitted prior noun phrase. This is consistent with the findings that human participants tend to slow down their reading times when encountering the critical verbs that disambiguate the syntactic structure of a garden-path sentence (e.g., Bever, 1970). The current result therefore suggests that the increased reading times on critical verbs may associate with lower predictability of the verbs given prior syntactic and semantic context. Yet intriguingly, the effect of thematic fit is no longer present when unambiguous sentences are being processed; in light of how the LSTM model performs in processing ambiguous sentences (i.e., the significant difference between the good- and bad-fit conditions), I see the absence of modulation of fit as an indication that the LSTM model is highly sensitive to prior linguistic context and the role of punctuations (e.g., commas) assigned by the context, which here is to signal the termination of an event structure. Following such logic, a putative account of the absence of an effect of fit here is that the disambiguation of semantic context should have happened on punctuations to a great extent.

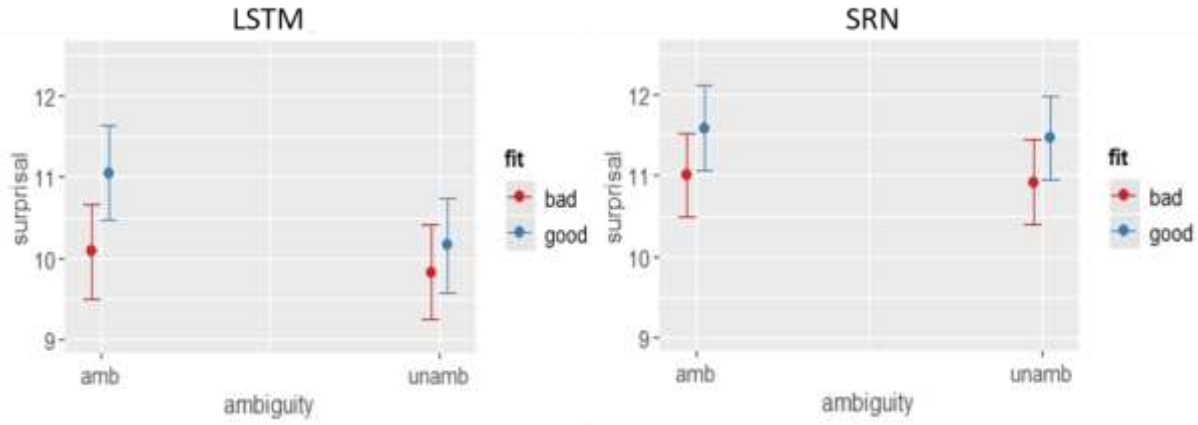

Figure 3: Effects of ambiguity and thematic fit on word surprisal. Left: LSTM; right: SRN

2.2 Model Comparison: SRN versus LSTM

As the effects included in the LMER model fitted to compare the output of the SRN and LSTM models could be too intricate for an elaborative description, in the text here I only highlight the effects which are most relevant to the contrast between the two network architectures. Full detail of the mixed model has been made available in Appendix (see **Appendix Table 2**).

First, LMER modelling shows a main effect of model (β= 1.0383, t=8.868, p<.001; here *lstm* is taken as zero reference), suggesting that surprisal estimates yielded by the SRN model is systematically higher than those yielded by the LSTM model. This is consistent with the findings in Part 1 that surprisal yielded by the LSTM has lower perplexity than that yielded by the SRN, as the LSTM

captures the statistical structures in language better than the SRN does, and thus has assigned lower surprisal to words.

Second, a significant interaction between model and fit is shown (β= -.7070, t= -4.232, p<.001), while the interaction between model and ambiguity shows no statistical significance (β= .1617, t= 0.976, p= .329542). Pairwise comparisons were also performed on these two interactions, but the significance level should be interpreted with extreme cautiousness as the p-values have not been adjusted, plus that a three-way interaction has also been introduced to the LMER model and has not been considered in the comparisons. Here I only compare the size of differences between conditions for the two RNN models according to t-ratios, without making claims about their significance level. Pairwise comparisons on model*fit suggest that the size of difference between the good- and bad-fit conditions is larger for the LSTM model (t = -2.786) than for the SRN model (t = -1.424). Pairwise comparisons on model*ambiguity yields similar results (LSTM: t = 5.836; SRN: t = 1.048). Therefore, I make a temporary (and possibly statistically unreliable) interpretation that the SRN model is less sensitive than the LSTM model in terms of both syntactic ambiguity and thematic fit.

Third and intriguingly, a significant three-way interaction between model and ambiguity and fit is observed (β= .6166, t= 2.610, p<.01), which means that the SRN and LSTM models show differing ways of interaction between ambiguity and fit. To help in comparison between the ways ambiguity and fit interact, a separate LMER model has been fitted to surprisal estimates of the SRN model and the effects of ambiguity and fit are visualised in the right panel of **Figure 3**. The mixed model on the SRN model output shows a significant main effect of thematic fit (β= .5822, t= 2.254, p=.02; here *bad* is taken as zero reference), indicating that the surprisal estimates of critical verbs made by the SRN model is higher when the thematic fit between the supposed subject of the critical verbs and its preceding verb is better. However, neither the main effect of ambiguity (β= -.09545, t= -1.271, p= .2164; here *amb* is taken as zero reference) nor the interaction between ambiguity and thematic fit (β= -.03748, t= -.379, p=.71) is shown significant.

Taken together, these results suggest that, different from those the LSTM model makes, the predictions the SRN model makes on the predictability of critical words are merely sensitive to the manipulation of semantic context (i.e., thematic fit) rather than syntactic ambiguity. Besides, the SRN model shows lower sensitivity to semantic context than the LSTM model does.

Part 3 General Conclusion and Discussion

In this assignment I investigated the quality of simple and gated RNNs as language models.

In the first part I examined the quality of the SRN and LSTM networks in terms of their abilities to 1) capture the statistical structures between words in English and 2) account for human sentence reading, and the development of the ability over training. First, I quantified model performance in capturing statistical relationships in language as perplexity, and compared perplexities across model type and training corpus size. I found that perplexity decreases steadily for both networks as the amount of training size accumulates, indicating that both simple and gated RNN networks performs better in capturing the statistical structures in language as they learn more sentences. Moreover, the SRN performs better than the LSTM when the training size is relatively smaller (i.e., between 1k-100k), but the advantage disappears and is even reversed at the end of training.

Second, using LMER modelling, I regressed human self-paced reading data against the output of each model (i.e., surprisal) at each training step to examine the development of the two networks in characterising human reading behavior. I quantified the goodness of fit of each model as the difference between model performance of a baseline model (without surprisal estimates) and a model with surprisal estimates yielded by each RNN model. The findings suggest that, similar to perplexities (in a converse manner though), goodness of fit of the SRN and LSTM models improved as the amount of training increases, and that the LSTM outperforms the SRN at the end of training. However, the current study also found that perplexities of the two models do not vary positively with goodness-of-fit. This suggests that that good capturing of statistical regularities in language does not (necessarily) equal higher predictive power in characterising human language processing. The results are consistent with numerous studies which have shown that RNN model output (i.e., surprisal) accounts for (at least some variance in) human reading data (e.g., Van Schijndel & Linzen, 2020; Aurnhammer & Frank, 2019), and that the model quality improves over training (Aurnhammer & Frank, 2019).

I acknowledge some limitations present in the Part 1. First, I compared model performance across type and training size, or correlates perplexities with goodness-of-fit by visual inspection rather than using statistical methods. This could have made the conclusions subject to bias from the observer (me). Second, the conclusion on RNNs derived in the current study was based only on one single training trajectory, which could have introduced particular variance in the output. Multiple training repetitions (e.g., altering the order of sentences to be learned by the same network architecture) are thus required in future studies to form a more generalisable conclusion about the development path of network performance.

In the second part I examined the ability of the LSTM and SRN networks to account for the garden-path phenomenon in sentence comprehension. I fitted a LMER model to investigate the LSTM model's performance in predicting the classic garden-path and its sensitivity to the modulation of semantic variations, i.e. the goodness of thematic fit between the subject of critical verbs and the first verb. I found that the LSTM model shows sensitivity to the modulation of both syntactic ambiguity and semantic variations (i.e. thematic fitness) and their interactions. To be specific, when processing ambiguous sentences, the LSTM model predicts surprisal as being significantly higher in critical verbs whose preceding noun phrase fits the previous semantic context badly than in those with a well-fitted prior noun phrase, which provides convergent evidence to Frank and Hoeks (2019). Intriguingly, however, the effect of thematic fit is no longer present when unambiguous sentences are being processed, which is different from the findings of Frank and Hoeks (2019) where the difference in surprisal between good and bad thematic-fit sentences is small yet present. Thus we failed to provide evidence that the LSTM makes the same predictions for the modulation of ambiguity and

thematic fit in English as in Dutch. My putative account of the absence of an effect of fit here is that the LSTM model is highly sensitive to prior linguistic context and the role of punctuations (e.g., commas) assigned by the context, and that the disambiguation of semantic context should have happened on punctuations to a great extent. To examine such an account, future research can compare surprisal estimates on the noun phrase immediately preceding the critical verbs (aka subject of critical verbs) in the four types of sentences. The expectation is that surprisal on the subject of critical verbs would be higher in syntactically unambiguous sentences than in ambiguous sentences regardless of thematic fit between the first verb and the subject of critical verbs, since punctuations would mark the start of a new sub-context.

I also fitted LMER models to compare the quality of the simple and gated RNN models in characterising the garden path effect. The results suggested that, different from those the LSTM model makes, the predictions the SRN model makes on the predictability of critical words were merely sensitive to the manipulation of semantic context (i.e., thematic fit) rather than syntactic ambiguity. Besides, the SRN model showed lower sensitivity to semantic context than the LSTM model does. In conclusion, compared to the SRN, the fully trained LSTM is better at characterizing the statistical structures in language, human sentence reading, and in particular garden-path effects. This suggests that gated RNN models such as the LSTM is able to learn (at least to some degree) syntactic and semantic relations in language simultaneously.

Reference

Aurnhammer, C., & Frank, S. L. (2018). *Comparing gated and simple recurrent neural network architectures as models of human sentence processing* [Preprint]. PsyArXiv. https://doi.org/10.31234/osf.io/wec74

Bever, T. (2013). *The cognitive basis for linguistic structures1*. https://doi.org/10.1093/ACPROF:OSO/9780199677139.003.0001

Frank, S. L., Fernandez Monsalve, I., Thompson, R. L., & Vigliocco, G. (2013). Reading time data for evaluating broad-coverage models of English sentence processing. *Behavior Research Methods*, *45*(4), 1182–1190. https://doi.org/10.3758/s13428-012-0313-y

Frank, S. L., & Hoeks, J. (2019). *The interaction between structure and meaning in sentence comprehension: Recurrent neural networks and reading times* [Preprint]. PsyArXiv. https://doi.org/10.31234/osf.io/mks5y

Schijndel, M. van, & Linzen, T. (2020). *Single-stage prediction models do not explain the magnitude of syntactic disambiguation difficulty*. PsyArXiv. https://doi.org/10.31234/osf.io/sgbqy

Appendix

Table 1: LMER model results for the surprisal estimates of the LSTM model

Fixed effects						
	Estimate	Std.	Error	df	t	Pr(>|t|)
(Intercept)	10.0825	0.2983	149.7952	33.798	< 0.0000000000000002	***
ambiguityunamb	-0.2532	0.1159	122.6794	-2.185	0.0308	*
fitgood	0.9728	0.3554	221.0579	2.737	0.006702	**
ambiguityunamb:fitgood	-0.6393	0.1653	122.6794	-3.867	0.000178	***
Random structure						
Groups	Name	Variance	Std.Dev.			
word	(Intercept)	5.4711	2.339			
word_pos	(Intercept)	0	0			
Residual	0.3894	0.624				

Table 2: LMER model results for comparison of the surprisal estimates of the SRN and LSTM models

Fixed effect						
	Estimate	Std. Error	df	t	Pr(>|t|)	
(Intercept)	10.0132	0.2782	166.7781	35.997	2.00E-16	***
ambiguityunamb	-0.2532	0.1171	346.1444	-2.16E+00	0.031293	*
fitgood	1.1353	0.3022	414.7344	3.757	0.000197	***
modelsrn	1.0383	0.1171	346.1444	8.868	2.00E-16	***
ambiguityunamb:fitgood	-0.6393	0.1671	346.1444	-3.827	0.000154	***
ambiguityunamb:modelsrn	0.1617	0.1656	346.1444	0.976	0.329542	
fitgood:modelsrn	-0.707	0.1671	346.1444	-4.232	2.97E-05	***
ambiguityunamb:fitgood:modelsrn	0.6166	0.2363	346.1444	2.61E+00	0.009453	**
Random Structure						
Groups	Name	Variance	Std.Dev.			
word	(Intercept)	5.2037	2.2812			
word_pos	(Intercept)	0	0			
Residual	0.3976	0.6305				