LING5702: Lecture Notes 15
Probabilistic parsing

We saw memory effects from vectors of neurons in an algorithmic-level sentence processing model.

Now we'll see expectation effects in reading from discrete structures at the computational level.

Contents

15.1 Structural surprisal model [van Schijndel et al., 2013] 1
15.2 Comparitor (neural net) transformer model [Vaswani et al., 2017] 3
15.3 Self-paced reading and eye-tracking data 4
15.4 Regression results of surprisal models [Oh et al., 2022] 5
15.5 Regression results of GPT-2 variants [Oh et al., 2022] 5

15.1 Structural surprisal model [van Schijndel et al., 2013]

Reading predictions come from **surprisal** – the log probability of each word given context:

$$\mathsf{S}(w_t \mid w_{1..t-1}) \stackrel{\text{def}}{=} -\log_2 \mathsf{P}(w_t \mid w_{1..t-1})$$

We can derive this from a structural model if we marginalize out the structure s_t:

$$-\log_2 \mathsf{P}(w_t \mid w_{1..t-1}) = -\log_2 \sum_{s_t} \mathsf{P}(w_t\ s_t \mid w_{1..t-1})$$

So instead of superposed vectors, we maintain lists of partial structures (derivation fragments).

Joint probabilities of words and structures are calculated from a recurrent sequence model:

$$\mathsf{P}(w_t\ s_t \mid w_{1..t-1}) \stackrel{\text{def}}{=} \sum_{s_{t-1}} \overbrace{\mathsf{P}(w_t\ s_t \mid s_{t-1})}^{\text{transition model}} \cdot \overbrace{\mathsf{P}(w_{t-1}\ s_{t-1} \mid w_{1..t-2})}^{\text{same joint at previous time step}}$$

where the transition model is broken into lexical and grammatical phases:

$$\begin{aligned}\mathsf{P}(w_t\ s_t \mid s_{t-1}) = \sum_{\ell_t, g_t} & \mathsf{P}(\ell_t \mid s_{t-1}) \cdot \\ & \mathsf{P}(w_t \mid s_{t-1}\ \ell_t) \cdot \\ & \mathsf{P}(g_t \mid s_{t-1}\ \ell_t\ w_t) \cdot \\ & \mathsf{P}(s_t \mid s_{t-1}\ \ell_t\ w_t\ g_t)\end{aligned}$$

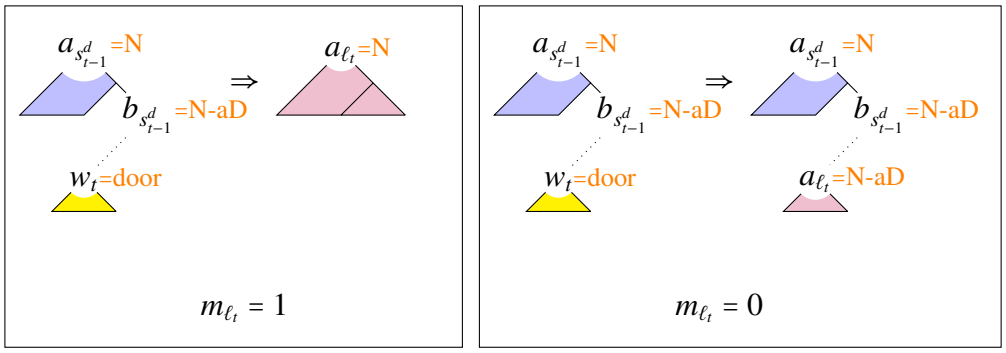
We'll need the expected frequency of category c as a left descendant ('left corner') of category c_0:

$$\mathsf{F}(c \mid c_0) \stackrel{\text{def}}{=} \overbrace{\sum_{n=1}^{N} \sum_{c_1,\ldots,c_n}}^{\text{for any path } c_1,\ldots,c_n \text{ of any length } n \text{ from 1 to } N} \overbrace{[\![c = c_n]\!] \prod_{i=1}^{n} \underbrace{\sum_{c'} \mathsf{P}(c_{i-1} \rightarrow c_i\ c' \mid c_{i-1})}_{\text{marginalize right child bc don't care}}}^{\text{probability of path from ancestor } c_0 \text{ to left descendant } c}$$

Then using $s_{t-1}=\langle a_{s^1_{t-1}}, b_{s^1_{t-1}}, \ldots, a_{s^D_{t-1}}, b_{s^D_{t-1}}\rangle$, define the probability of:

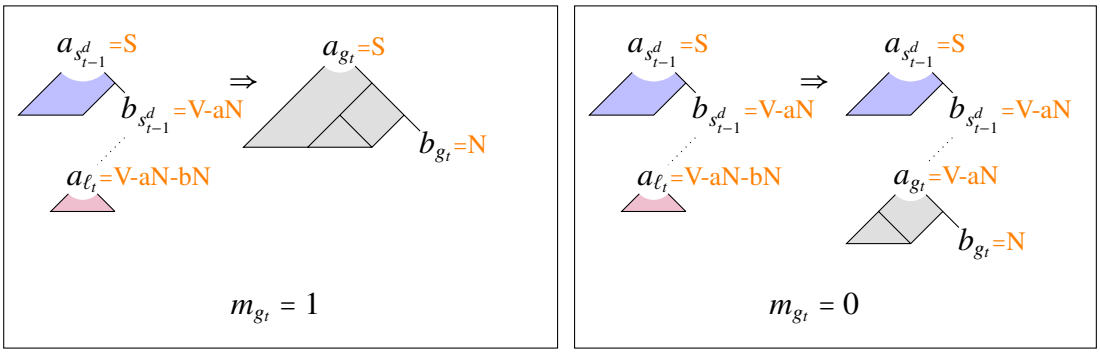
1. **lexical** (terminal) decisions $\ell_t=\langle m_{\ell_t}, a_{\ell_t}\rangle$ where $d = \text{argmax}_{d'}\{a^{d'}_{t-1}\neq\bot\}$:

$$\mathsf{P}(\ell_t \mid s_{t-1}) \stackrel{\text{def}}{=} \begin{cases} [\![a_{\ell_t}=a_{s^d_{t-1}}]\!] \cdot \overbrace{\frac{[\![b_{s^d_{t-1}}=b_{s^d_{t-1}}]\!]}{[\![b_{s^d_{t-1}}=b_{s^d_{t-1}}]\!]+\mathsf{F}(b_{s^d_{t-1}}|b_{s^d_{t-1}})}}^{\text{probability of zero-length path}} \cdot \overbrace{\mathsf{P}(b_{s^d_{t-1}} \to w_t \mid b_{s^d_{t-1}})}^{\text{probability of terminal}} & \text{if } m_{\ell_t}=1 \\ \underbrace{\frac{\mathsf{F}(a_{\ell_t}|b_{s^d_{t-1}})}{[\![a_{\ell_t}=b_{s^d_{t-1}}]\!]+\mathsf{F}(a_{\ell_t}|b_{s^d_{t-1}})}}_{\text{probability of longer path}} \cdot \underbrace{\mathsf{P}(a_{\ell_t} \to w_t \mid a_{\ell_t})}_{\text{probability of terminal}} & \text{if } m_{\ell_t}=0 \end{cases}$$



2. **grammatical** (non-terminal) decisions $g_t=\langle m_{g_t}, a_{g_t}, b_{g_t}\rangle$ where $d = \text{argmax}_{d'}\{a^{d'}_{t-1}\neq\bot\} - m_{\ell_t}$:

$$\mathsf{P}(g_t \mid s_{t-1}\, \ell_t\, w_t) \stackrel{\text{def}}{=} \begin{cases} [\![a_{g_t}=a_{s^d_{t-1}}]\!] \cdot \overbrace{\frac{[\![b_{s^d_{t-1}}=b_{s^d_{t-1}}]\!]}{[\![b_{s^d_{t-1}}=b_{s^d_{t-1}}]\!]+\mathsf{F}(b_{s^d_{t-1}}|b_{s^d_{t-1}})}}^{\text{probability of zero-length path}} \cdot \overbrace{\mathsf{P}(b_{s^d_{t-1}} \to a_{\ell_t}\, b_{g_t} \mid b_{s^d_{t-1}})}^{\text{probability of non-terminal}} & \text{if } m_{g_t}=1 \\ \underbrace{\frac{\mathsf{F}(a_{g_t}|b_{s^d_{t-1}})}{[\![a_{g_t}=b_{s^d_{t-1}}]\!]+\mathsf{F}(a_{g_t}|b_{s^d_{t-1}})}}_{\text{probability of longer path}} \cdot \underbrace{\mathsf{P}(a_{g_t} \to a_{\ell_t}\, b_{g_t} \mid a_{g_t})}_{\text{probability of non-terminal}} & \text{if } m_{g_t}=0 \end{cases}$$



Finally, define probability of $s_t=\langle a_{s^1_t}, b_{s^1_t}, \ldots, a_{s^D_t}, b_{s^D_t}\rangle$ where $d = \text{argmax}_{d'}\{a_{s^{d'}_{t-1}}\neq\bot\} + 1 - m_{\ell_t} - m_{g_t}$:

$$\mathsf{P}(s_t \mid s_{t-1}\, \ell_t\, w_t\, g_t) \stackrel{\text{def}}{=} \prod_{d'=1}^{D} \begin{cases} [\![a_{s^{d'}_t}, b_{s^{d'}_t} = a_{s^{d'}_{t-1}}, b_{s^{d'}_{t-1}}]\!] & \text{if } d'<d \\ [\![a_{s^{d'}_t}, b_{s^{d'}_t} = a_{g_t}, b_{g_t}]\!] & \text{if } d'=d \\ [\![a_{s^{d'}_t}, b_{s^{d'}_t} = \bot, \bot]\!] & \text{if } d'>d \end{cases}$$

15.2 Comparitor (neural net) transformer model [Vaswani et al., 2017]

Transformers associate 'queries' and 'keys' of K items to choose targets of attention.

These associations are modeled using 'query', 'key' and 'value' matrices $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{D \times D}$.

Each item in a transformer is represented in a D-dimensional vector $\mathbf{H}_\ell \in \mathbb{R}^{D \times K}$ at each level ℓ.

At each level, each item may 'attend' to one other item per 'head' h.

This is done by comparing queries and keys, using inner products of these as a similarity measure.

Values, weighted by this similarity, are then passed to the next level:

$$\mathbf{H}_{\ell,h} = \overbrace{\mathbf{V}_{\ell,h}\,\mathbf{H}_{\ell-1}}^{\text{value for each target}}\ \mathrm{SoftMax}\Big(\underbrace{\big(\overbrace{\mathbf{K}_{\ell,h}\,\mathbf{H}_{\ell-1}}^{\text{key for each target}}\big)^\top \overbrace{\mathbf{Q}_{\ell,h}\,\mathbf{H}_{\ell-1}}^{\text{query for each source}}}_{\text{attention matrix}}\Big)$$

where SoftMax is our multinomial logistic function on $\mathbf{M} \in \mathbb{R}^{J \times N}$ with N instances of J values:

$$\mathrm{SoftMax}(\mathbf{M}) = \frac{\exp(\mathbf{M})}{\mathbf{1}^\top \exp(\mathbf{M})}$$

The outputs $\mathbf{H}_{\ell,h}$ of the heads are then concatenated and fed into another (e.g. sigmoid) layer FF:

$$\mathbf{H}_\ell = \mathrm{FF}\Big(\underbrace{\sum_h \delta_h \otimes \mathbf{H}_{\ell,h}}_{\text{concatenate}}\Big)$$

Run the model with several different words to calculate $\mathsf{P}(w_t \mid w_1, \ldots, w_{t-1})$.

Experiments used several surprisal models as comparitors:

- *vSLC* [van Schijndel et al., 2013]: A left-corner parser based on a PCFG with subcategorized syntactic categories [Petrov et al., 2006], trained on a generalized categorial grammar reannotation of Sections 02 to 21 of the WSJ corpus.
- *Structural* [Oh et al., 2022]: Same but extended with semantic contexts and morphology.
- *JLC* [Jin & Schuler, 2020]: A neural left-corner parser based on stack LSTMs [Dyer et al., 2015], trained on Sections 02 to 21 of the WSJ corpus.
- *RNNG* [Hale et al., 2018, Dyer et al., 2016]: An LSTM-based model with explicit phrase structure, trained on Sections 02 to 21 of the WSJ corpus.
- *GPT2XL* [Radford et al., 2019]: GPT-2 XL, a 48-layer decoder-only autoregressive Transformer model trained on ~8B tokens of the WebText dataset.
- *5-gram* [Heafield et al., 2013]: A 5-gram language model with modified Kneser-Ney smoothing trained on ~3B tokens of the English Gigaword Corpus [Parker et al., 2009].

- *GLSTM* [Gulordava et al., 2018]: A two-layer LSTM model trained on ~80M tokens of the English Wikipedia.
- *JLSTM* [Jozefowicz et al., 2016]: A two-layer LSTM model with CNN character inputs trained on ~800M tokens of the One Billion Word Benchmark [Chelba et al., 2013].

15.3 Self-paced reading and eye-tracking data

Structural and comparitor surprisal models were fit to reading time observations from:

1. Self-paced reading times from 181 subjects – 10 naturalistic stories: 10,245 tokens.

   The data were filtered to exclude observations corresponding to sentence-initial and sentence-final words, observations from subjects who answered fewer than four comprehension questions correctly, and observations with durations shorter than 100 ms or longer than 3000 ms.

   This resulted in a total of 770,102 observations

   All observations were log-transformed prior to model fitting.

2. Eye-gaze durations from 10 subjects – 67 newspaper editorials: 51,501 tokens.

   The data were filtered to exclude unfixated words, words following saccades longer than four words, and words at starts and ends of sentences, screens, documents, and lines.

   This resulted in a total of 195,507 observations

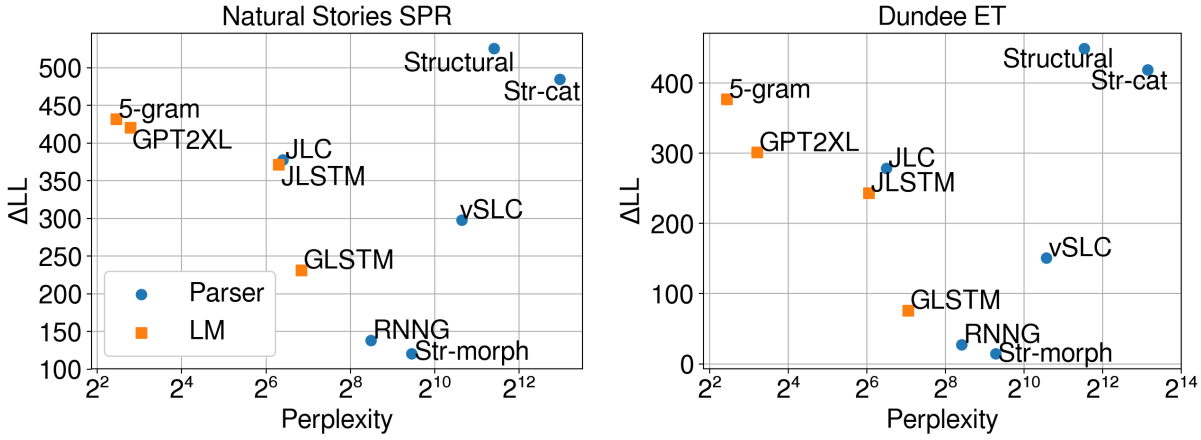
   All observations were log-transformed prior to model fitting.

Linear regressions were fit with the following baseline predictors, both with and without surprisal:

- Self-paced reading times [Futrell et al., 2021]: word length measured in characters, index of word position within each sentence
- Eye-gaze durations [Kennedy et al., 2003]: word length measured in characters, index of word position within each sentence, saccade length, whether or not the previous word was fixated

15.4 Regression results of surprisal models [Oh et al., 2022]

Structural models predict words better (perplexity is $\frac{1}{P(w_t|w_1,\ldots,w_{t-1})}$), but not reading time:



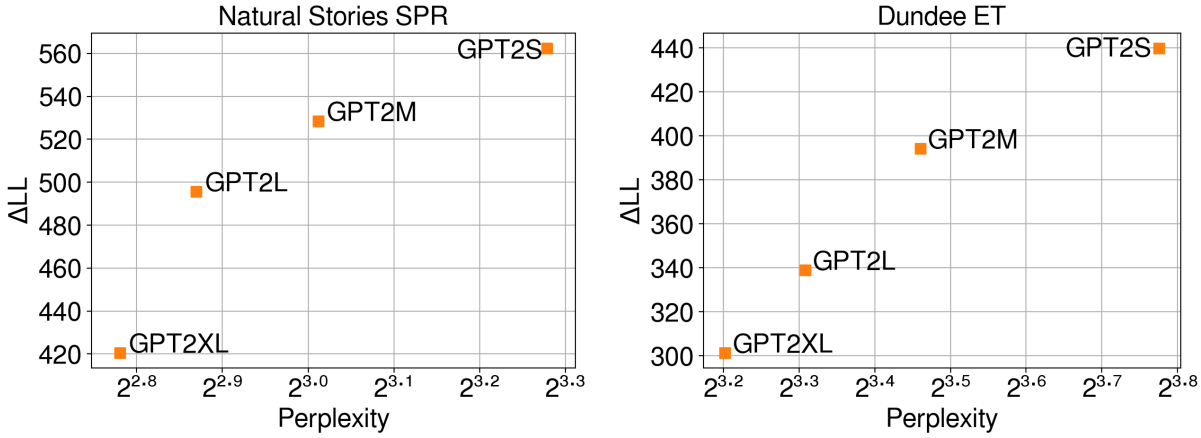
LMER on SPR durations: baseline LL: -18988.9 (log probability of fit w. Gaussian noise)

LMER on go-past times: baseline LL: -64927.3

15.5 Regression results of GPT-2 variants [Oh et al., 2022]

GPT-2 (transformer model) predicts reading times worse as models get larger:

- *GPT2S*: GPT-2 Small, which has 12 layers and ~124M parameters;
- *GPT2M*: GPT-2 Medium, which has 24 layers and ~355M parameters;
- *GPT2L*: GPT-2 Large, which has 36 layers and ~774M parameters;
- *GPT2XL*: GPT-2 XL, which has 48 layers and ~1558M parameters.



LMER on duration; Data: [Futrell et al., 2021]; Baseline LL: -18988.9

LMER on go-past; Data: [Kennedy et al., 2003]; Baseline LL: -64927.3

References

[Chelba et al., 2013] Chelba, C., Mikolov, T., Schuster, M., Ge, Q., Brants, T., Koehn, P., & Robinson, T. (2013). One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*.

[Dyer et al., 2015] Dyer, C., Ballesteros, M., Ling, W., Matthews, A., & Smith, N. A. (2015). Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers* (pp. 334–343).: The Association for Computer Linguistics.

[Dyer et al., 2016] Dyer, C., Kuncoro, A., Ballesteros, M., & Smith, N. A. (2016). Recurrent neural network grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 199–209).

[Futrell et al., 2021] Futrell, R., Gibson, E., Tily, H. J., Blank, I., Vishnevetsky, A., Piantadosi, S., & Fedorenko, E. (2021). The Natural Stories corpus: A reading-time corpus of English texts containing rare syntactic constructions. *Language Resources and Evaluation*, 55, 63–77.

[Gulordava et al., 2018] Gulordava, K., Bojanowski, P., Grave, E., Linzen, T., & Baroni, M. (2018). Colorless green recurrent networks dream hierarchically. In *NAACL-HLT* (pp. 1195–1205).

[Hale et al., 2018] Hale, J., Dyer, C., Kuncoro, A., & Brennan, J. (2018). Finding syntax in human encephalography with beam search. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics* (pp. 2727–2736).

[Heafield et al., 2013] Heafield, K., Pouzyrevsky, I., Clark, J. H., & Koehn, P. (2013). Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics* (pp. 690–696). Sofia, Bulgaria.

[Jin & Schuler, 2020] Jin, L. & Schuler, W. (2020). Memory-bounded neural incremental parsing for psycholinguistic prediction. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies* (pp. 48–61).

[Jozefowicz et al., 2016] Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., & Wu, Y. (2016). Exploring the limits of language modeling. *CoRR*.

[Kennedy et al., 2003] Kennedy, A., Pynte, J., & Hill, R. (2003). The Dundee corpus. In *Proceedings of the 12th European conference on eye movement*.

[Oh et al., 2022] Oh, B.-D., Clark, C., & Schuler, W. (2022). Comparison of structural parsers and neural language models as surprisal estimators. *Frontiers in Artificial Intelligence*, 5.

[Parker et al., 2009] Parker, R., Graff, D., Kong, J., Chen, K., & Maeda, K. (2009). *English Gigaword LDC2009T13*.

[Petrov et al., 2006] Petrov, S., Barrett, L., Thibaux, R., & Klein, D. (2006). Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics (COLING/ACL'06)*.

[Radford et al., 2019] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *ArXiv*.

[van Schijndel et al., 2013] van Schijndel, M., Exley, A., & Schuler, W. (2013). A model of language processing as hierarchic sequential prediction. *Topics in Cognitive Science*, 5(3), 522–540.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In *NIPS* (pp. 5998–6008).