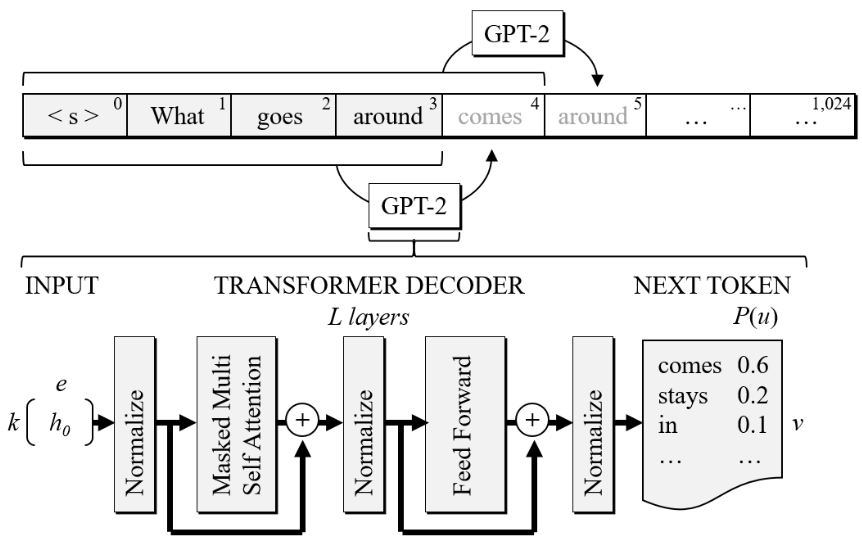
1.1 GPT-2 Model Description

To provide the intuition behind transformer-based NLG models, we briefly illustrate the mechanics of the popular GPT-2 model. Given a sequence of tokens with context window size k, $U=(u_{-k},\dots,u_{-1})$, the objective of the autoregressive model GPT-2 is to accurately "predict" the next likely word[1] (Figure W1.1) by sampling from a probability distribution over its entire learned vocabulary (consisting of 50,257 tokens) conditional on the given word sequence and on a pre-trained neural network with parameters Θ. Model pre-training tries to maximize the likelihood in equation (W1) for an unsupervised corpus of words ($\mathcal{U}$) (Radford et al. 2018).

Figure W1.1: The GPT-2 Model[2]



$$L_1(\mathcal{U}) = \sum_i logP(u_i|u_{i-k},\dots,u_{i-1};\Theta) \quad \text{(W1)}$$

$$h_0 = UW_e + W_p \quad \text{(W2)}$$

---

[1] For ease of discussion, we describe the model in terms of "words." GPT-2 is derived using BPE (Byte Pair Encoding) and tokens (i.e., learned and encoded pieces of words).

[2] Visualization derived from Radford et al. (2018), and adapted to depict the updated GPT-2 architecture.

$$h_l = transformer\_block_{(h_{l-1})} \forall i \in [1, L] \quad \text{(W3)}$$

$$P(u) = softmax(h_L W_e^T) \quad \text{(W4)}$$

In essence, GPT-2 relies on word and given context meaning information to generate its output distribution over its vocabulary. More specifically, the data input consists of a matrix h_0 (W2), where the given word sequence U, word meaning information in terms of word embeddings W_e, and sequential word position information in terms of position embeddings W_p are combined. As illustrated in Figure W1.1, information from h_0 is extracted, transformed, added and normalized multiple times (to ease processing), and projected into the embedding space e by L layers of decoder transformer blocks (W3). This information includes the extent of putting attention on a given word sequence using multi-headed self attention (Vaswani et al. 2017), and high dimensional hidden language states to shift the focus in the embedding space e to recreate natural word sequences from position-wise feed forward neural networks. The output of the final block h_L projects all this information into the embedding space and is multiplied with GPT-2's original (unconditional) transposed word embeddings matrix W_e^T to assess which word from the GPT-2 vocabulary best matches the information contained in h_L (W4). The multiplication of h_L and W_e^T can be thought of as a similarity or matching between the embedding space distribution of the output of h_L (containing meaning, position, attention, and hidden language states information) and the unconditional embedding space distribution of each respective vocabulary word. More similarity of a vocabulary word in terms of its embedding to h_L will result in a higher probability in GPT-2's output distribution. GPT-2 then obtains a probability distribution over its vocabulary $P(u)$ (W4) and can sample the upcoming word in the sequence from the most likely words in $P(u)$.

Using the above procedure, GPT-2 learned and stored word probabilities for given word sequences represented in its 345 million parameters (including embeddings, attention weight matrices and Θ) using 8 million English text documents with a broad topical variety. Neural network parameters Θ were first initialized and then trained on batches of 512 sequences. The loss function refers to the language modeling cross entropy loss, where 1 is assigned to the word that appears next (u_i) in the training sequence (e.g., “comes” in Figure W1.1), and 0 to all other words in GPT-2’s vocabulary, and compare the log transformed GPT-2 softmaxed output probability value P_u for that respective word to appear next. A loss of 0 means the GPT-2 prediction was in perfect accordance with the actual next word (i.e., 1), the higher the deviation of the GPT-2 prediction (e.g., 1-0.6 = 0.4 for “comes” in Figure W1.1) to the actual word, the more the loss value increases. During training, GPT-2 performs this process on batches and mini-batches of several sequences before updating Θ.

Appendix References

Baayen RH, Shafaei-Bajestan E (2019) Analyzing linguistic data: A practical introduction to statistics. Package 'languageR'. Version 1.5.0. *CRAN*. Accessed May 20, 2019, https://cran.r-project.org/web/packages/languageR/languageR.pdf

Benoit K, Watanabe K, Wang H, Nulty P, Obeng A, Müller S, Matsuo A, (2018) "quanteda: An R package for the quantitative analysis of textual data." *Journal of Open Source Software*. 3(30). https://doi.org/10.21105/joss.00774

Berger J, Sherman G, Ungar L (2020b) TextAnalyzer. Accessed November 11, 2020, http://textanalyzer.org

Bronnenberg BJ, Kim JB, Mela CF (2016) Zooming in on choice: How do consumers search for cameras online? *Marketing Science*. 35(5):693-712.

Danaher PJ, Mullarkey GW, Essegaier S (2006) Factors affecting website visit duration: A cross-domain analysis. *Journal of Marketing Research*. 43(2):182-194.

Edelman B, Zhenyu L (2016) Design of search engine services: Channel interdependence in search engine results. *Journal of Marketing Research*. 53(6):881-900.

Flanigan, AJ, Metzger, MJ (2007) The role of site features, user attribtues, and information verification behaviors on the perceived credibility of web-based information. *New Media & Society*. 9(2):319-342. https://doi.org/10.1177/1461444807075015

Jerath K, Ma L, Park YH (2014) Consumer click behavior at a search engine: The role of keyword popularity. *Journal of Marketing Research*. 51(4):480-486.

Kamoen N, Holleman B, Bergh H (2013) Positive, negative, and bipolar questions: The effect of question polarity on ratings of text readability. *Survey Research Methods*. 7(3):181-189.

Liu J, Toubia O (2018) A semantic approach for estimating consumer content preferences from online search queries. *Marketing Science*. 37(6):930-952.

Maechler M, Rousseeuw P, Croux C, Todorov V, Ruckstuhl A, Salibian-Barrera M, Verbeke T, Koller M, Conceicao ELT, Palma MA (2020) Basic robust statistics. Package 'robustbase'. Version 0.93-6. *CRAN*. Accessed May 20, 2020, https://cran.r-project.org/web/packages/robustbase/robustbase.pdf

Pennebaker JW, Booth RJ, Boyd RL, Francis ME (2015) Linguistic inquiry and word count: LIWC2015. Austin, TX: Pennebaker Conglomerates. Accessed November 1, 2020, www.LIWC.net.

Pitler E, Nenkova A (2008) Revisiting Readability: A unified framework for predicting text quality. *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. 186-195.

Radford A, Narasimhan K, Salimans T, Sutskever I (2018) Improving language understanding by generative pre-training. OpenAI.

Roberts C (2010) Correlations among variables in message and messenger credibility scales. *American Behavioral Scientist*. 54(1):43-56.

Rocklage MD, Rucker DD, Nordgren LF (2018) Persuasion, emotion and language: the intent to persuade transforms language via emotionality. *Psychological Science*. 29(5):749-760.

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomze AN, Kaiser L, Polosukhin I (2017) Attention is all you need. *31st Conference on Neural Information Processing Systems (NIPS 2017)*. 1-15.