Proposing Word Saliency Methods for Strategic Masking during the Training of Non-Autoregressive Logical Data-to-Text Generation Models

Yale University Department of Statistics & Data Science

December, 2021

Luke Benson
luke.benson@yale.edu

Abstract

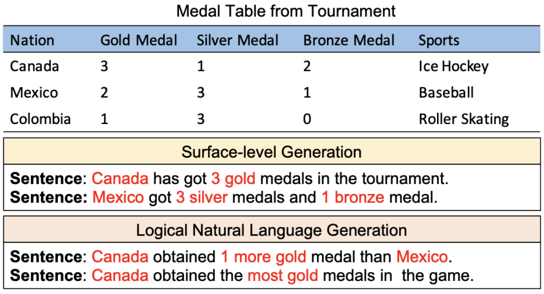
Logical data-to-text (D2T) models generate summarization sentences which are logically-entailed by the information contained within a table. Current models utilize sequential generation schemes in which sentences are produced unidirectionally from left to right. In this paper, we apply the mask-predict algorithm to generate sentences non-sequentially with the intent of addressing some of the major issues faced by these sequential natural language generation (NLG) schemes. Through this process, we also develop a novel measure of word saliency which can be considered for various NLG tasks. While our models do not produce results which come close to state-of-the-art, our approach to the task of logical D2T generation may serve as springboard for future research into alternative logical D2T model training strategies.

1 Introduction

Data-to-text generation (D2T) is an emerging task within the field of natural language generation (NLG). Given a structured representation of data, such as a table, D2T models are trained to produce human-readable text which communicate some component of the information contained within the table. There are a myriad of real-world applications of these models, including generating weather forecasts, sports game summaries, and business reports.

When evaluating the quality of machine-generated sentences, there are two aspects which must be considered: fluency and fidelity. *Fluency* refers to the sentence's grammatical correctness, while *fidelity* measures the factuality of the sentence given the underlying information. Large-scale transformer-based language models such as BERT (Devlin et al., 2019) and GPT-3 (Brown et al., 2020) are capable of generating highly fluent sentences for a variety of tasks, but they are not trained to write sentences that are necessarily factually based in the data. Thus, several recent papers published within the field of NLG have sought to build D2T models which generate sentences that are both fluent and faithful to the underlying data (Liu et al., 2018; Puduppully et al., 2019). While these models show promising results for accomplishing both tasks, they succeed only in producing *surface-level* generations, which means that the generated sentences simply restate individual facts contained within the data.

A recent paper by Chen et al. (2020) proposed a collection of D2T models which are capable of producing logically-entailed sentences: summarizations that communicate logical operations over several data points. These sentences go beyond surface-level summarizations and more closely resemble the kinds of advanced conclusions that humans can draw from data. **Figure 1** is used in the Chen et al. paper to illustrate the difference between the surface-level and logical generations.



Medal Table from Tournament

Nation	Gold Medal	Silver Medal	Bronze Medal	Sports
Canada	3	1	2	Ice Hockey
Mexico	2	3	1	Baseball
Colombia	1	3	0	Roller Skating

Surface-level Generation

Sentence: Canada has got 3 gold medals in the tournament.
Sentence: Mexico got 3 silver medals and 1 bronze medal.

Logical Natural Language Generation

Sentence: Canada obtained 1 more gold medal than Mexico.
Sentence: Canada obtained the most gold medals in the game.

Figure 1: Distinction between surface-level and logical generations (Chen et al., 2020).

As **Figure 1** demonstrates, a model which learns surface-level summarizations is able to communicate the fact that Canada won 3 gold medals or that Mexico won 2 gold medals. Logical D2T models are able to take this information and go one step further. Canada has 3 gold medals and Mexico has 2, which means that Canada won 1 more gold medal than Mexico.

The contribution of this project is a proposal for an alternative training scheme for logical D2T generation. Current state-of-the-art logical D2T models utilize *autoregressive* generation techniques: given the data contained within the table, text in a sentence is produced sequentially from left to right. The generation of each word is, thus, conditioned on previously-generated words. To address some of the challenges that arise from sequential D2T generation, we assess the fluency and fidelity of sentences which are produced through a *non-autoregressive* generation scheme. While there are various non-autoregressive algorithms, the overarching idea is that words in a sentence are generated simultaneously before the sentence as a whole is then modified. With the intent of improving upon the initial summarizations produced by the standard non-autoregressive algorithm, we then inject information regarding a word's importance or informativeness – i.e. *word saliency* – into the training of our models.

2 Background & Relevant Work

2.1 Autoregressive Logical D2T Generation

D2T models operate within the typical encoder-decoder architecture of neural NLG models, where data from the table is encoded into a lower-dimensional latent space and then a decoder translates that latent representation into text. In their baseline models, Chen et al. (2020) utilize autoregressive decoder techniques for their text generation. As a result of producing text word-by-word, however, these autoregressive decoders have the potential to write sentences which are not logically entailed by the data contained within the table. Each generated word is unaware of the words that will appear after it, and so there may be beginnings of sentences for which there are no logical ends. **Figure 2** is presented in the Chen et al. paper to highlight this challenge.

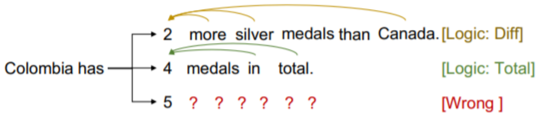


Figure 2: Illustrating the primary challenge for autoregressive generation techniques (Chen et al., 2020).

Figure 2 highlights three possible sentences generated through autoregressive decoding. All of the sentences begin with "Colombia has", and the remaining words in each sentence are conditioned on this initial text. When "2" and "4" are the next words generated, there are logical conclusions to the sentences: Colombia has 2 more silver medals than Canada and 4 medals in total. However, there is no relationship for which "5" is significant to Colombia in the table. Thus, when "5" is the next word generated, the model will ultimately write an illogical statement. Since, in this case, the numbers in these sentences describe the relationship between entities, their generation should be dependent on those entities. Ideally, "2" would not be dependent on "Colombia has", but rather "Colombia", "Canada", and "silver medals".

2.2 Non-Autoregressive Decoding

Within the field of machine translation, modelers have begun to employ non-autoregressive decoding techniques through which words in a sentence are generated in parallel (Gu et al., 2018; Gu et al., 2019). Since all words are initially written simultaneously, non-autoregressive decoders, unlike their autoregressive counterparts, do not inherently create sequential dependencies. Sequential dependencies create syntactic structure within a sentence, and so sentences generated non-autoregressively do have the potential to be less grammatically correct. However, a significant possible benefit of applying non-autoregressive decoding techniques to logical D2T generation is reducing the frequency at which illogical sentences are produced. In the third sentence from **Figure 2**, for example, the generation of "5" would no longer solely be conditioned on "Colombia has".

Mask-predict (Ghazvininejad et al., 2019) is one such non-autoregressive decoding scheme. This algorithm takes the encoded representation of data contained within the table and first determines the length of the output sequence. Each word in that output sequence is then generated in parallel and assigned a probability based on the latent representation. The sequence positions containing words that have the lowest probabilities – i.e. those that the model has the least confidence in – are then "masked," or turned back into positions which contain unknown words. Conditioned on both the latent representation and the currently-unmasked output words, new words are then generated for those masked positions. This process repeats over a preset number of iterations, with the number of masked words decreasing linearly at each step until a final output sentence is predicted.

2.3 Word Saliency

The concept of *saliency* in machine learning refers to the importance or informativeness of the components of an input. It is a widely-researched topic within the field of computer vision, as researchers attempt to train neural models which can identify the most salient objects or regions within images (Hou and Zhang, 2007; Yang et al., 2013; Liu and Han, 2016). Understanding which parts of an image are the most salient can assist in tasks such as image compression and content-based image retrieval. Word saliency methods, which are currently less well-researched, seek to identify the most important or informative words within a body of text. There are two distinct approaches to measuring word saliency: task-agnostic and task-specific. *Task-agnostic* algorithms determine a word's saliency through the degree centrality of the word within a graph representation of the entire text corpus; the task for which the text is used is irrelevant to words' scores (Mihalcea and Tarau, 2004; Xu et al., 2019). *Task-specific* algorithms, on the other hand, score words based on the change in the probability of a given task after erasing or masking each word from the text (Li et al., 2017; Ren et al., 2019).

Word saliency scores obtained through a fluency-specific algorithm may be useful to a given NLG task because they would provide additional knowledge regarding a word's importance to the grammatical correctness of a statement. In the case of non-autoregressive logical D2T generation, accounting for a word's value to the fluency of text may help counteract the possible reduction in fluency brought forth by non-autoregressive generation.

3 Problem Description & Methodology

3.1 Problem Description

Our contribution to the task of logical D2T generation is the proposal for a word saliency-infused mask-predict model training scheme. While this training strategy is distinct from the various strategies put forth by Chen et al. (2020), the mathematical problem ultimately remains the same. We also test our training strategy using the data set designed by Chen et al., LogicNLG ($\sim$28k sentences). This data set is *open-domain*, meaning that the input tables are not confined to a particular subject, but rather span a wide variety of topics. Thus, the model we train on this data set will be applicable to any set of tables.

Within the LogicNLG data set, we are given a set of input tables. Each table T has n rows and p columns, where T_{ij} denotes the value – i.e. number, word, phrase – contained within the ith column and jth row. For each table, we are also given a collection of sentences which are logically-entailed by the data from the table. Each sentence Y is made up of a collection of words, $Y = (y_1, y_2, \ldots y_n)$. Given an input table T and an output sentence Y, we aim to train a D2T model which learns to maximize the probability of the sentence given the table, $P(Y|T)$. The training algorithm should be designed to further encourage the model to generate sentences which are both fluent and faithful to the underlying data. The specific mechanics of our saliency-infused non-autoregressive training scheme are outlined in the following section.

3.2 Methodology

There are two distinct steps in creating natural text summarizations from data in a table: encoding and decoding. In the encoding phase, we take the same approach as Chen et al. (2020). We *linearize* the table into one large natural language statement using a pre-trained language model. Thus, our table can now be thought of as a single paragraph which tells us which values are contained in each row and column.

3.2.1 Standard Non-Autogressive Decoding Scheme

In the decoding phase, we begin with a baseline mask-predict algorithm as proposed by Ghazvininejad et al. (2019). Based on the latent representation of data contained in the table, we predict the length of the output sentence, n, and the initial words in the sentence, $y_1, y_2, \ldots y_n$:

$$y_i^0 = \arg\max_w (P(y_i = w|x)$$

The probability associated with each word is then simply the probability of the chosen word conditional on the latent representation:

$$p_i^0 = \max_w (P(y_i = w|x)$$

At each iteration of the algorithm, we then mask the k words that have the lowest probabilities, $y_{mask}^t = \arg\min_i(p_i, k)$, and generate new words for those k positions based on the latent representation and the remaining unmasked words:

$$y_i^t = \arg\max_w(P(y_i = w|x, y_{unmasked}^t)$$

$$p_i^t = \max_w(P(y_i = w|x, y_{unmasked}^t)$$

At each step, t, the number of masked words, k, is simply a function of the sequence length, n, and the total number training iterations, T:

$$k = n * \frac{(T - t)}{T}$$

We train our decoder over $T = 5$ iterations.

Our goal is to develop an effective logical D2T model for the LogicNLG data set. Since we are developing all our model parameters and hyperparameters from scratch, we first pre-train our model on another data set, ToTTo, before continuing to train and refine our parameters on LogicNLG. ToTTo is a larger open-domain D2T data set ($\sim$120k sentences) developed by Parikh et al. (2020). Like LogicNLG, the summarization sentences in ToTTo also utilize logical reasoning that go beyond surface-level summarizations.

3.2.2 Saliency-Infused Non-Autogressive Decoding Scheme

There is a distinction in how the mask-predict algorithm operates during the model training process and during the actual generation process. When the decoder masks words during summary generation, words are masked through the process outlined above. In the training stage of our standard mask-predict decoder, words are chosen to be masked at random. In this way, the model learns broadly how to assess which words it should be confident in during sentence generation. However, we hypothesize that masking words strategically during training may improve the performance of our baseline non-autoregressive model. Given the potential decrease in fluency of our standard non-autoregressive generations, we may want to modify the training process to prioritize grammatical correctness. If we mask the words that are most important or informative to the fluency of the sentence, our model may learn to predict more informative words effectively whenever it is given less informative words.

For example, let us consider a scenario in which we are generating a sentence and 80% of the words in the output sequence are currently masked. The remaining 20% of words are the ones that the model was most confident in with regard to the latent representation of the data, but they may not be very informative of the sentence's actual grammatical structure. In this scenario, our proposed model may be better able to generate the remaining, more informative words. Thus, to address this hypothesis, we seek to incorporate a measure of each word's salience to a sentence into our non-autoregressive D2T training.

We propose a task-specific measure of word saliency based on the process of word erasure and calculated using GPT-2. GPT-2, the predecessor to GPT-3, is a transformer-based language model which has been trained to predict the next word given all the previous words within some text (Radford et al., 2019). It was trained on text from eight million internet pages and has more than 1.5 billion model parameters. Given the immense size of the model and the scope of the data on which it was trained, GPT-2 is capable of completing NLG tasks beyond word prediction, such as question answering, reading comprehension, and translation. Although its performance on these additional tasks is far from state-of-the-art, GPT-2 can perform adequately without any task-specific training.

For any given sentence, GPT-2 is able to assign a probability score which indicates how likely that exact sentence is to occur. Given the extensiveness of the training of GPT-2, we ground our measure of word salience in this score. Our algorithm takes in a given sentence and first calculates the probability of this sentence. We then remove each word one-by-one and calculate the percentage change in the sentence probability when each word is erased. The resulting saliency score for each word is precisely this calculated percentage change. This process attempts to capture how informative a word is to the fluency of a sentence. More salient words are those whose erasure impacts the probability of the sentence the greatest.

During the training of our saliency-infused mask-predict model, we begin the decoding process the same way as before: by predicting the output sequence length and the word at each position of the output sequence. Rather than masking words at random, however, we mask the words which are most salient to the current predicted output se-

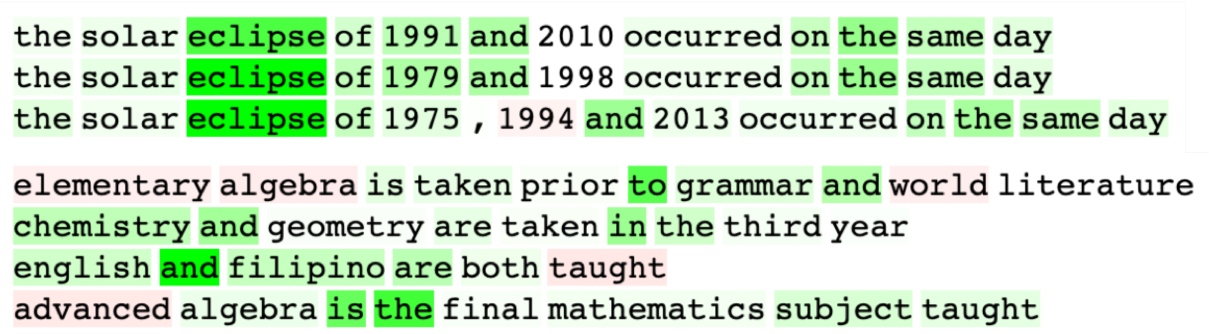

the solar eclipse of 1991 and 2010 occurred on the same day
the solar eclipse of 1979 and 1998 occurred on the same day
the solar eclipse of 1975 , 1994 and 2013 occurred on the same day
elementary algebra is taken prior to grammar and world literature
chemistry and geometry are taken in the third year
english and filipino are both taught
advanced algebra is the final mathematics subject taught

Figure 3: Word saliency scores as determined by GPT-2 for example sentences in the LogicNLG training set.

quence. At each iteration, we have a new sequence of words, and so the salience of a word that remains unmasked may change over time. The goal of this training strategy is to help the model learn how to generate high-saliency words when it is given low-saliency words during sentence generation.

4 Results

The analysis of our results is split into two parts: 1) an investigation into the scores produced by our proposed measure of word saliency, and 2) an assessment of the fluency and fidelity of sentences generated through our proposed D2T training scheme on LogicNLG.

4.1 Word Saliency Scores

We can first investigate the efficacy of our saliency scoring algorithm through a kind of case study by observing its performance on specific sentences within the LogicNLG training set. **Figure 3** exhibits how the scoring algorithm operates on two groupings of sentences within this training set. The color and intensity of the highlighting of each word indicates how salient that word is: more salient words have a darker shading of green. Red highlighting signifies that the word has a negative salience score, which means that the probability of the sentence as assessed by GPT-2 actually improves when the word is erased.

In the first grouping, “eclipse” has the highest saliency score across all three sentences, indicating that the removal of this word leads to the greatest drop in the probability of the sentence. This finding is in line with our expectations, given that “the solar of…” is grammatically illogical. Saliency scores speak to words’ syntactical weight in this way, and so they also simultaneously shed light on words’ expendability. “1994” has a negative salience score because “the solar eclipse of 1975 and 2013…” is actually a more likely statement than “the solar eclipse of 1975, 1994 and 2013…”. While both are grammatically correct, the latter sentence contains a greater level of detail, and so is naturally less to occur within a piece of text, The same phenomena appears in the second grouping of sentences in the case of “elementary”, “world”, and “advanced”.

We can also examine the results of the saliency scorer on a larger scale. Within the training sets of both ToTTo and LogicNLG, we calculate word saliency for every word in every sentence. For each word, we then average its saliency scores across all sentences to arrive at an average word saliency within each data set. **Figure 4** shows the most and least salient frequently-appearing words within the LogicNLG training set.

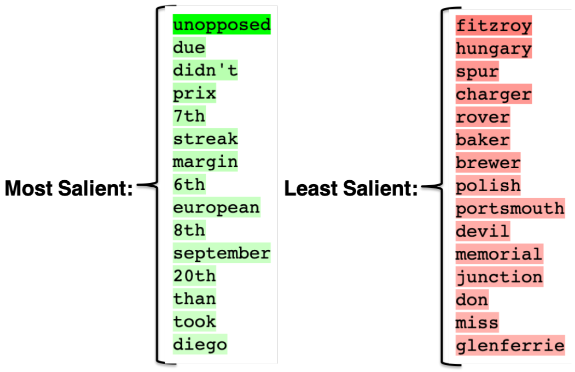


Figure 4: The most and least salient frequent words (25+ instances throughout ∼28k sentences) for the LogicNLG training set.

Through qualitative evaluations of these scores, we can see that many of the least salient words are named entities directly contained within the data set (e.g. “hungary”, “charger”, “rover”). Some of the most salient words are also named entities from the input tables (e.g. ”prix”, ”european”), Many of the most salient words, however, either reflect the relationships between entities (e.g. “unopposed”,

Decoding Tecnhnique	SACREBLEU	ROUGE-L	PARENT	BERTScore	NLI-Acc	SP-Acc
Autoregressive	15.40	33.99	26.25	87.75	69.69	41.02
Standard Non-Autoregressive	12.00	31.66	20.30	86.69	45.41	39.19
Saliency-Infused Non-Autoregressive	10.10	30.04	19.85	86.22	39.35	37.35

Table 1: Fluency and fidelity metric results for autoregressive, standard non-autoregressive, and saliency-infused non-autoregressive generations.

"7th", "6th") or link phrases (e.g. "due", "than").

It is difficult to explain why many of the least salient words are named entities. Named entity words are mostly nouns, and we would generally anticipate that the placement of a noun would be crucial to the fluency of a sentence. It is possible that these words are assigned low or negative salience scores when they are seen in an unfamiliar context. "Spur", for example, refers exclusively to the San Antonio Spurs basketball team in the LogicNLG training sentences. Since we do not usually refer to a spur as a noun with agency, it is possible that a sentence excluding "spur" is more probable when the spur is alluded to as an agent in the original sentence. Then again, we would expect that GPT-2 is well-trained enough to recognize the context of this word.

The observation that many salient words reflect relationships between entities or link phrases aligns with the fact that word saliency is measured through the effect of word erasure. Without words to describe the relationship between two or more entities, the sentence is not only illogical, but incoherent. Similarly, words linking phrases or parts of sentences are crucial to the interpretability of a sentence.

4.2 D2T Generations

We evaluate our model generations using a collection of fluency and fidelity metrics. For fluency, we utilize SACREBLEU (Post, 2018), ROUGE-L (Lin, 2004), PARENT (Dhingra et al., 2019), and BERTScore (Zhang et al., 2020). To measure fidelity, we utilize the NLI-Acc and SP-Acc metrics proposed by Chen et al. (2020) in the LogicNLG paper. These fidelity metrics are far from perfect, but there is also less extensive research regarding NLG fidelity metrics in general.

Both our standard and saliency-infused non-autoregressive decoding strategies are pre-trained on ToTTo. **Table 1** contains the fluency and fidelity metrics results for the standard non-autoregressive scheme, the saliency-infused non-autoregressive scheme, and a similarly-trained simple autoregressive scheme on the LogicNLG test set.

Our standard non-autoregressive model trained through the typical mask-predict algorithm performs comparably, but consistently worse than the basic autoregressive decoder model across all fluency and fidelity metrics. The saliency-infused non-autoregressive model performs even slightly worse across the board.

It is certainly possible that either additional model training or more extensive pre-training would lead to improvements in the fluency and fidelity scores. In the case of the standard mask-predict decoder, we anticipate that effective training would eventually allow us to achieve close to state-of-the-art results, given that non-autoregressive techniques have been able to do so for the task of machine translation. However, there is a more plausible explanation beyond simply additional model training for why the saliency-infused mask-predict algorithm leads to a reduction in performance. In saliency-infused training, unlike standard training, we are ultimately restricting our masking procedure to follow a particular pattern. The model consistently leaves lower-salience words unmasked, and so it learns only how to predict high-saliency words given low-saliency words. It is, thus, prevented from seeing a diverse set of scenarios during the training process and fails to learn how to generate low-saliency words when it is left with high-saliency words. This presents a problem for sentence generation, as the model needs to know how to generate all words given any other set of words.

5 Conclusion

The contribution of this project is two-fold:

1) We propose a measure of word saliency based on the process of word erasure and calculated using GPT-2.

2) We incorporate the saliency scores into an investigation of an alternative training scheme for the task of non-autoregressive logical D2T generation.

While our proposed training scheme fails to achieve results which are close to state-of-the-art, the concept of word saliency could be useful when applied through an alternative training method. In future research, for example, we can test the effectiveness of a training curriculum in which the model first learns to generate low-saliency words when given high-saliency words, and only then is trained to predict high-saliency words from low-saliency ones. Both 1) additional model training and 2) more extensive pre-training may also be needed to fully evaluate the potential of standard and saliency-infused non-autoregressive methods for this task.

6 References

Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T.J., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., & Amodei, D. (2020). Language Models are Few-Shot Learners. ArXiv, abs/2005.14165.

Chen, W., Chen, J., Su, Y., Chen, Z., & Wang, W.Y. (2020). Logical Natural Language Generation from Open-Domain Tables. ACL.

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL.

Dhingra, B., Faruqui, M., Parikh, A.P., Chang, M., Das, D., & Cohen, W.W. (2019). Handling Divergent Reference Texts when Evaluating Table-to-Text Generation. ArXiv, abs/1906.01081.

Ghazvininejad, M., Levy, O., Liu, Y., & Zettlemoyer, L. (2019). Mask-Predict: Parallel Decoding of Conditional Masked Language Models. EMNLP.

Gu, J., Bradbury, J., Xiong, C., Li, V.O., & Socher, R. (2018). Non-Autoregressive Neural Machine Translation. ArXiv, abs/1711.02281.

Gu, J., Wang, C., & Zhao, J. (2019). Levenshtein Transformer. NeurIPS.

Hou, X., & Zhang, L. (2007). Saliency Detection: A Spectral Residual Approach. 2007 IEEE Conference on Computer Vision and Pattern Recognition, 1-8.

Li, J., Monroe, W., & Jurafsky, D. (2016). Understanding Neural Networks through Representation Erasure. ArXiv, abs/1612.08220.

Lin, C. (2004). ROUGE: A Package for Automatic Evaluation of Summaries. ACL 2004.

Liu, N., & Han, J. (2016). DHSNet: Deep Hierarchical Saliency Network for Salient Object Detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 678-686.

Liu, T., Wang, K., Sha, L., Chang, B., & Sui, Z. (2018). Table-to-text Generation by Structure-aware Seq2seq Learning. ArXiv, abs/1711.09724.

Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing Order into Text. EMNLP.

Parikh, A.P., Wang, X., Gehrmann, S., Faruqui, M., Dhingra, B., Yang, D., & Das, D. (2020). ToTTo: A Controlled Table-To-Text Generation Dataset. ArXiv, abs/2004.14373.

Post, M. (2018). A Call for Clarity in Reporting BLEU Scores. WMT.

Puduppully, R., Dong, L., & Lapata, M. (2019). Data-to-Text Generation with Content Selection and Planning. ArXiv, abs/1809.00582.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners. OpenAI Blog.

Ren, S., Deng, Y., He, K., & Che, W. (2019). Generating Natural Language Adversarial Examples through Probability Weighted Word Saliency. ACL.

Xu, S., Li, H., Yuan, P., Wu, Y., He, X., & Zhou, B. (2020). Self-Attention Guided Copy Mechanism for Abstractive Summarization. ACL.

Yang, C., Zhang, L., Lu, H., Ruan, X., & Yang, M. (2013). Saliency Detection via Graph-Based Manifold Ranking. 2013 IEEE Conference on Computer Vision and Pattern Recognition, 3166-3173.

Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., & Artzi, Y. (2020). BERTScore: Evaluating Text Generation with BERT. ArXiv, abs/1904.09675.

7 Appendix

All code and data pertaining to the development of word saliency scores and model outputs can be found at https://github.com/lukegbenson/saliency-based-NAR-D2T.