Controlled text generation for Political Speeches using Improved Plug and Play approach

Madhu Samhitha V mvangara@umass.edu
Somya Goel somyagoel@umass.edu
Madhumitha Mohan madhumithamo@umass.edu
Haritha Ananthakrishnan hananthakris@umass.edu

1 Problem statement

In the political realm, speeches have always been the most important tool for creating maximum impact and engagement. A well-delivered speech can help a speaker gain the trust of the public and direct their campaigns much more effectively. Recent advancements in the field of Language Modeling have achieved unprecedented results in the task of text generation. While these Language Models enhance grammatical correctness and coherence in general, there have also been recent studies focus on controlling properties of the text and steering the text in a particular direction. In our project, we develop a text generator specifically for political speeches and control attributes like political affiliation, and the topic of interest. We train a baseline model by fine-tuning GPT-2 on parliamentary speeches, and develop two other models based on Plug and Play Architecture (Dathathri et al., 2020). This architecture allows users to easily add or delete attribute models to steer text in different direction, without the need to fine-tune large Language Models. Our motivation was to build a handy tool, which would aid speakers get their vision across to the masses in an efficient and impactful manner.

2 What you proposed vs. what you accomplished

- We collected the dataset of the parliamentary speeches from the years 1890 -2011 and performed various joins to obtain an exhaustive list of features and performed feature selection with respect to our use case as proposed in our proposal.
- Our Baseline model is fine-tuned GPT-2 on the Parliamentary data described above.
- We built baseline models with subsets of data, and evaluated their performances
- We built a PPLM model, with attribute models for Politics, Sub-topic and Political Affiliation, and evaluated their performances
- We built another model to improve performance, by combining fine-tuning and PPLM architecture and obtained better results
- Additional attribute models for persuasion and sentiment analysis were proposed in the proposal, but were not implemented. Adding multiple attribute models decreased the semantic performance of LMs, and the results reacted less to all attributes. In the scope of this project, we chose a subset of important attributes. Also, the dataset for persuasive speeches was unbalanced, and was not suitable for our problem.
- Human evaluation is by far the gold standard for Text generation problems. We conducted a detailed evaluation study, by defining multiple metrics and collected evaluations from multiple human evaluators. We then analysed these results and compared performances across models, and identified particular cases where models performed the best and poorly.
- This was followed by performing the Error analysis.

3 Related work

(Tang et al., 2019) had proposed a method to generate target-guided conversations, which acted as the foundation for our project. (Niu and Bansal, 2018) created personality-based conversational agents focusing on the politeness of re-

sponses, and (Peng et al., 2018),in his paper enabled controlling of the ending of stories. (Kassarnig, 2016) presents a system that can generate speeches for a desired political party using a language model for grammatical consistency and a topic model for textual context. The paper uses a simple statistical language model based on n-grams, specifically a 6-gram model. For the topic model, a POS tag pattern filter (ex: Noun Conjunction Noun) for 2 and 3-gram terms is used.

Several approaches have been explored to control text generation like fine-tuning models with Reinforcement Learning , (Ziegler et al., 2019) training conditional generative models (Hatori et al., 2018); (Ficler and Goldberg, 2017), and training language models with control codes. (**?**) propose the Plug and Play Model (PPLM) which takes an alternative approach to generate controlled text by plugging in multiple smaller attribute models which steer the pre-trained language models in the desired direction.

In our attempt to make the speech more stimulating we studied the research around effective Natural Language generation. (Piwek and van Deemter, 2003) describes Natural language generation mainly focusing on the degree to which the generated text can influence and tap into the emotional appeal of the listeners. The speech generated should encompass a strong inclination of a set of morals, judgments and biases. This characterization of the stance the speech takes is called the notion of affect. Persuasive NLG is closely related to affective NLG in the lines of understanding the different lexical aspects of the speakers' style, inclination, and sentiments.

(Gentzkow and Taddy, 2020) has given the physical set of all the speeches and debates made in the Parliament from the years 1873-2011. This dataset became the foundation of this project and was used to train all the model included in the project. (**?**) suggested the use of human evaluation as a way of evaluating results instead of relying on the usual non-human evaluation measures. GPT-2 is a widely used model to perform text generation that gives good results as seen in this paper. (Brown et al., 2020)

4 Your dataset

Congressional Records dataset contains processed text from the bound and daily editions of the United States Congressional Record, as provided by HeinOnline. For our dataset, we used the bound edition which covers the 43rd to 111th Congresses. Each edition includes all text spoken on the floor of each chamber of Congress: the United States House of Representatives and the United States Senate. As in the Figure 1 This corpus contains six groups of information

- Speech text files, counts of stemmed bigrams, and speech-level metadata from each session of Congress
- Vocabulary from all sessions of Congress
- a subset of the vocabulary manually classified into 22 substantive topics (finance, healthcare, abortion laws, etc)
- the most partisan phrases from each
- the names of the political parties represented in each session of congress
- Speaker Metadata

4.1 Data Explanation

The structure of the dataset is as follows:

- SpeakerMap43.txt - SpeakerMap111.txt : Contains **Speaker Metadata**. This contains information such as chamber, first name, last name, gender, and party of the speaker
- Descr43.txt - Descr111.txt : Contains **Speech Metadata**. This contains important information that we will be filtering our dataset based on.
  - **Date of speech** (Ranges from 1873 to 2011)
  - **Speaker** (How the speaker is referred to in the speech)
  - **Character count** - Char count of the speech
  - **Word count** - Word count of the speech
- Speeches43.txt - Speeches111.txt - The actual list of **speeches** of that particular session of Congress.

4.1.1 Feature Selection and Filtering

- **Word count** - Since we used GPT-2 and set a 1024 subword token limit, we decided to choose full speeches that had word count between **250 to 300** words

Table 1: Format of Speech Files

File	Content	Key	Format
byparty_2gram_###.txt	bigram counts by party	party, phrase	2-column
byspeaker_2gram_###.txt	bigram counts by congressperson	speakerid, phrase	3-column
descr_###.txt	speech metadata	speech_id	14-column
speeches_###.txt	full-text speeches	speech_id	2-column
###_SpeakerMap.txt	speaker metadata	speech_id	10-column

Figure 1: Dataset Description

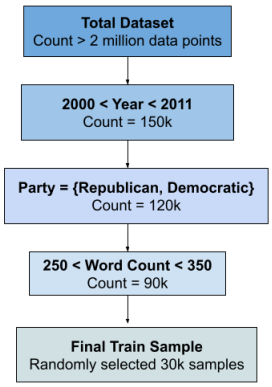

Figure 2: Dataset Filtering Basis

- **Year of Speech** - Our dataset contains speeches from 1873-2011. Since we do not want our model to learn ancient ideologies we limit the year of given speech starting year 2000 onwards.
- **Partisanship** : Our dataset had the following ratio of parties
  - **Republican** : 46%
  - **Democrat** : 43%
  - **Other Parties** : 11%

  In order to limit our models to a biparty standpoint, we filtered for only speeches given by members of the Republican and Democratic parties. Figure 2 shows a visual representation of the counts of various data points

4.1.2 Topic Based Seperation

The topics.txt file also contains most commonly occuring bigrams of the following 22 topics : **Alcohol, government, budget, health, business, immigration, crime, justice, defense, labor, economy, mail, education, minorities, elections, money, environment, religion, federalism, tax, foreign, and trade**

All this information and code can be found in the GitHub repo under "Data_Preprocessing.py"

5 Baselines

For our baseline model, we use the fine-tuning approach, where different models are fine-tuned on subsets of the Congressional speeches, for different attribute values. For instance, in our evaluation, we initially compare speeches generated for the sub-topic "business" under politics, generated by our different models. Here, our baseline model is gpt-2 fine-tuned on the subset of political speeches from the topic "business".

The generated speeches from the baseline model are coherent to politics and the sub-topic, better than the models implemented in our main approach. But this method involves fine-tuning different models for different combinations of attributes, and would be a tedious / resource-intensive method especially if the number of control attributes increase. Thus we propose two approaches using PPLM to expedite this process and increase flexibility. From our filtered dataset of 90k speeches, we chose a random sample of 30k speeches as the training data. For validation and test, we took subsets of text from the remaining speeches, and used the first 20 tokens as the

prompt. For testing, we also used a manually created some prompts not related to parliamentary speeches, to analyse performances on different prompts. The hyper-parameters chosen are batch_size=16, epochs=5, lr=2e-5.

6 Your approach

We build our approaches on top of the Plug and Play Model (PPLM) architecture (Dathathri et al., 2020), which enables the steering of text according to multiple attributes, eliminating the necessity to fine-tune models according to each possible attribute. Given an attribute "a", to steer the model towards this attribute while preserving grammar, we shift the history H_t in the direction of the sum of two gradients: one toward higher log-likelihood (LL) of the attribute "a" under the conditional attribute model $p(a \mid x)$ and one toward higher LL of the unmodified language model p(x).

Additionally, to ensure the fluency of GPT-2 and avoid adversarial examples, the latents are updated to minimize the Kullback–Leibler (KL) Divergence between outputs of the conditional and unconditional LM. The PPLM architecture uses GPT-2 as the pre-trained Language Model (LM). From the combined model, we follow the approximate Metropolis-adjusted Langevin (MALA) sampler by following gradients in the latent representation space as done by (Nguyen et al., 2017). This model provides users the flexibility to add attributes to state-of-the-art LMs easily without fine-tuning. We ran our models with Colab Pro, with 25GB RAM and GPU.

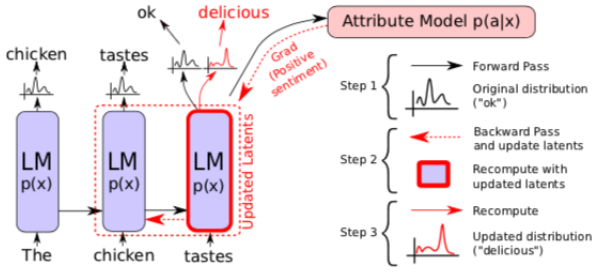

Figure 3: Plug and Play Architecture

6.1 Approach 1

With PPLM, we experiment two approaches to generate political speeches. In approach 1, as illustrated in Figure 4, we use three attribute models, first to choose the main topic as politics, second to choose the sub-topic within politics (like Business, Healthcare, etc.) and the third to control the political affiliation of the speaker (Democratic / Republican). To choose the sub-topic we used a simple BoW approach, based on keywords for topics present in the dataset. To implement the third political affiliation model, we trained a classifier on top of our pre-trained Language Model, which is GPT-2. We use a total of 30k speeches, consisting of a stratified balanced sample of Democratic and Republican texts. The hyper-parameters chosen were batch size = 32 and number of epochs = 5. The file "party_discriminator" contains the code used to implement this.

We used PPLM's code as reference to train our generator, with the help of libraries like Hugging Face Transformers and PyTorch. The code used for approach 1 is attached as "approach1" in our repository. The code first generates an unperturbed text from GPT-2, and then steers this along specified attributes according to the attribute models.

While steering, in an attempt to preserve the grammatical correctness and semantic meanings of GPT-2, we try to minimize the KL divergence between the outputs of the modified and unmodified texts. We use the hyper-parameter of KL-loss co-efficient as 0.01. The architecture also allows us to specify the step_size, which controls the intensity of attribute model. From our validation results, we observed that for higher step_size, the model is steered more towards attributes, but at the cost of losing semantic and grammatical meanings. So choose a step_size of 0.005 after experimenting with different values.

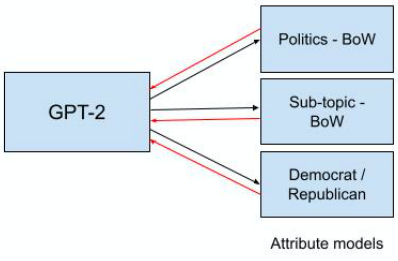

Figure 4: PPLM - Approach 1

6.2 Approach 2

The results from approach 1, though fairly coherent, they sounded less like speeches and more like

a narrated text. The grammatical correctness, relevance and coherence is also compromised. To improve this, we used another approach illustrated in Figure 5 combining the advantages of fine-tuning and PPLM, where GPT-2 was first fine-tuned with the parliamentary speeches. We then control this text with Plug and Play architecture, by adding two attribute models on top of the fine-tuned model, one for sub-topic BoW, and the other to control political affiliation. The political affiliation classifier here, is built on top of the fine-tuned parliamentary speeches LM as well, unlike Approach 1 where we used just GPT-2 as classifier pre-trained model. The code for approach 2 is given in the file "approach2" in the GitHub repo.

A detailed evaluation of both these approaches, and comparison with the baseline model is provided in the error analysis section. We notice that this model produces results which are more relevant, semantically better and coherent. Here, only a single fine-tuned model with speeches is required, unlike the baseline approach where different models for different attribute values are required. This combined approach improves the results of approach 1 significantly, as the model is now better suitable for political speeches, when compared to the BoW method in approach 1 to steer towards politics. We also perform hyper-parameter tuning and use step size = 0.005 and KL-loss co-efficient as 0.01. We initially generate results using a single attribute model with BoW, and then add another attribute model, the political affiliation classifier.

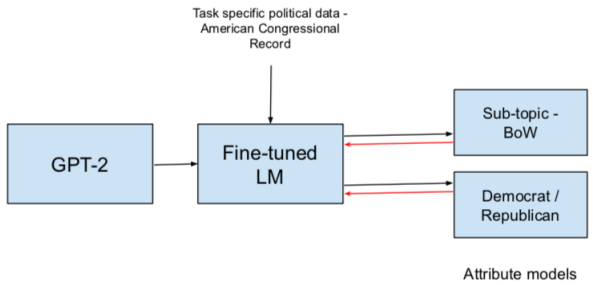

Figure 5: PPLM - Approach 2

7 Evaluation of our models

After literature survey, and incorporating the feedback, We have Natural language generation (NLG), a sub-field of natural language processing (NLP), deals with building software models that produce readable and coherent (Celikyilmaz et al., 2021) NLG is commonly considered a general term which encompasses a wide range of tasks that take a form of input (e.g., a structured input like a dataset or a table, a natural language prompt or even an image) and output a sequence of text. Given the myriad of ways in we any human can interpret text, Human evaluation is considered as the gold standard for such systems. The qualitative and quantitative analysis was conducted.

7.1 Evaluation Guidelines

We provided our annotators with a brief description of the project and acquainted them with the problem statements and the expectations. The annotators were chosen after careful consideration of their domain expertise and familiarity and interests in political speeches and fluency in the English Language. A total of 5 annotators were chosen. For the process of Human evaluation of our project, we sampled 30 prompts from Heinbound Congressional dataset from the test data and made sure that the model was not familiar with these set of prompts. We applied various filters and hand picked the prompts which were an inclusive representation of the data. Various filtering attributes included the length of the prompt, the political flavour, and quality of English. This marked the completion of the **prompt subset selection**. We used a token length limit of 100 for the speech generation.

Models subset selection for Annotation Our model building stages included a deep dive exploration of various speech generation models utilising the capabilities of GPT-2 with the different PPLM models to choose from.

The models chosen for the evaluation studies as a part of our experiments-

- Baseline model: GPT-2 fine tuned on the Dataset curated from Section. 4
- PPLM model from Approach 1[Refer Figure 7] with Business BOW Attribute model
- PPLM Model from Approach 2[Refer Figure 7] with Business BOW Attribute model
- PPLM Model from Approach 1 with Economics BOW Attribute model and Party Discriminator

- PPLM Model from Approach 2 with Economics BOW Attribute model and Party Discriminator

7.2 Evaluation metrics and guidelines

After careful selection of the evaluation metrics and guidelines as described below, the evaluators were asked to grade on a scale of [1-5].

- **Relevance** - Is the text that is generated relevant to a political speech in a parliamentary setting. Is it a speech format and not a reported speech or an article format. The speech must be of format of engaging an audience in the political setting. For Models with additional Attribute models, The relevance has to be graded in mind that the text generated should also contain a flavour of democratic speaker on the lines of economics.

- **Non Redundancy** - The text that is being generated should not be repetitive in nature and consist of sentences that are unique. Please make sure that occurring sentences are not similar forms of each other.

- **Coherence** - Are the sentences that are being formed make sense in a cohesive manner? Does the whole speech that is generated read out in a seamless manner? How do you grade the readability ?

- **Consistency** - The speech generated revolves around one topic or does it wander off to topics not consistent with the prompt given? Does it contradict itself in the end?

- **Correctness** - Please evaluate the grammatical construction of the generated text.

8 Results and Analysis

The following were the observations from our evaluation studies - The mean of the scores annotated was calculated for each model across different annotators, we can clearly see that the Approach 2 exhibits a significant improvement over the baseline model and Approach 1. This can be attributed to the fact that Approach 2 exploits the benefits of fine-tuning while increasing flexibility with PPLM. We also observe that even though our baseline example was fine-tuned with sub-topic samples, the PPLM BoW approach, steers text by including relevant words, making our annotators give higher scores for relevance.

8.1 Qualitative Error analysis

Prompt - I thank the gentleman for yielding. madam speaker. i rise in strong support of h.r. 4626. the health insurance industry

Text generated by our best model (This one is our personal favourite!) This received a the maximum 5's for all the metrics by all our annotators : I thank the gentleman for yielding. madam speaker. i rise in strong support of h.r. 4626. the health insurance industry. i also support the bipartisan legislation that is currently in the committee markup. it would provide subsidies to low-income families. it would provide funding to the states to ensure that low-income families are eligible to participate in health insurance programs that benefit low income families and that are available for low income families. i urge the leadership of both houses of congress to proceed with this bill and to act immediately. the health care industry is committed to supporting my colleague houston speaker.

We observed that our baseline does not perform when it comes to short prompts which are not so verbose. In our Approach 2, We can see that our model is able to able to perform way better with a mean Relevance metric of 4.48. For example -
Prompt - I want to start by congratulating Laurel on her victory tonight

Baseline Model produces : Sam continued. "I have known Laurel for many years. She was my personal coach for years. We went to meet some of our best competitors. Her delivery was fantastic. I was fortunate enough to have been a close friend of her father. We have known each other for decades. I always want to thank her for this chance to go to an Ivy League school."

PPLM with Approach 1 produces : [I want to start by congratulating Laurel on her victory tonight. She did a great job as a leader of the team. She was very focused and determined in how she conducted herself. I think she showed her true colors. I was proud to work with her for this team and I hope she keeps this team together.¡Continued¿]
Analysis: We can observe from the above examples the texts generated from a very generic

model	coherence_mean	relevance_mean	non_red_mean	consistency_mean	correctness_mean
baseline	3.59	3.67	3.81	3.26	3.68
approach2_business	4.48	4.55	4.53	4.38	4.41
approach1_business	3.22	2.91	3.17	3.10	3.54

Figure 6: Results from both the Approaches

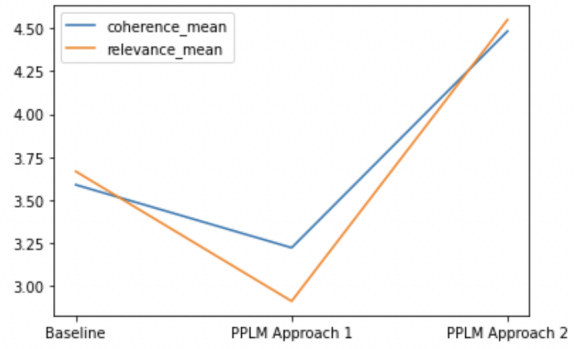

Figure 7: Analysis of Relevance and Coherence across various models

short prompt failed to stay relevant in the political setting even though it remains grammatically correct.

PPLM with Approach 2 produces: ['I want to start by congratulating Laurel on her victory tonight. She did a great job tonight and was the first lady to have the honor of serving in the U.S. House and Senate. She has been a terrific ambassador of goodwill to the world. She was a leader in the administration that has helped build communities and communities that are thriving right now. She has been a champion of working with the world community to improve their health and security and to help the country heal and prosper"

Analysis: We can clearly see how our best performing model is steer the text generation towards in the US political environment.

Prompt - What a day to be alive

Generated Text What a day to be alive.Im a woman of color who has lived in the United States for 10 years. Iv́e worked my entire life in the industry of photography. I was born on the first day that a friend introduced me to the concept of post-racial photography and Iv́e been doing that for almost two decades now. The most recent post of mine on the internet has been titled "The Black Body."

Analysis: Even for a shorter prompt with almost no reference to any political keywords is able to steer the text generation towards staying relevant, coherent and consistent. In a lot of such instances we could observe that our model even though was trained on the data which encompassed parliamentary speeches from a date range of [2000-2010]. Our model by utilising the advantages that come with of GPT-2 and the diversity of PPLM is able to write a better introduction to the speech which is extremely relevant to today's society and ideologies.

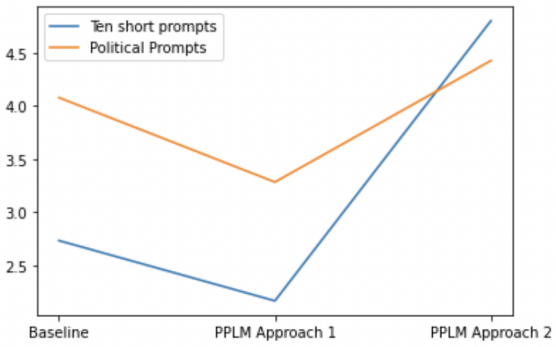

Figure 8: Analysis of Relevance scores for short vs verbose political prompts

8.2 Quantitative Error Analysis

For quantitative analysis, we perform ablation study by evaluating the performance fine-tuned baseline, adding one BoW attribute model, and then adding another classifier attribute model. The results obtained are discussed below:

Figure 7 shows the plot of coherence and relevance scores across different models. We see that Approach 1 behaves poorly when compared to the baseline, but Approach 2 improves significantly.

We then analyse the results, based on the prompts provided. We observe that for political verbose prompts, the values generated are much higher than short generic prompts. So we plot two line charts in Figure 8 across our three models. We

observe that the political prompts produce more relevant results than the short prompts on an average. In Approach 2, the short prompt performs better. The evaluation metrics generated show incremental improvements in consistency and relevance.

To evaluate the performance of the model on two attribute models together, we generated text from topic "politics", sub-topic "economics" and political affiliation "democrat". We show the results for "relevance" obtained for these, using step sizes 0.01 and 0.005 in Figure 9. In the figure, we observe that step size 0.01 decreases the performance of both the approaches in general. We also observe that the redundancy as shown in Figure 10 significantly increases with increase in step size.

While comparing approaches, we observe that results corresponding to Approach 2 are significantly better than Approach 1, in line with the results reported above.

Approach	StepSize = 0.005	StepSize = 0.01
Approach 1	3.333333	2.933333
Approach 2	4.333333	3.366667

Figure 9: Relevance scores for different step sizes

Approach	StepSize = 0.005	StepSize = 0.01
Approach 1	3.833333	1.700000
Approach 2	3.666667	3.033333

Figure 10: Redundancy scores for different step sizes

9 Contributions of group members

Majority of our work towards this project was done in a collaborative manner, and all 4 of us were deeply involved with tasks related to this project. We collaborated on the tasks from the beginning of the project cycle till its end. Some of these tasks include, performing the initial literature survey, gathering the requirements, doing the initial setup, implementations of models, deciding on hyperparameters etc. But still, we are mentioning the below contributions depending on the maximum work done by the teammate towards that particular task.

- Madhu Samhitha V : Designing the evaluation guidelines and the selection of annotators, deciding the evaluation metrics, performing the qualitative and quantitative analysis with visualisations.
- Somya Goel : Performing the literature survey to decide on various elements of the model, identifying the future scope of the project and performing the intermediate tasks so that the whole workflow of the project remained seamless.
- Madhumitha Mohan : Designing and implementing the PPLM model - Approach 1, and improved performance by incorporating fine-tuning to PPLM in Approach 2.
- Haritha Ananthakrishnan : Implementing the baseline model (fine-tuned GPT model), along with the required data preprocessing required to run the models.

10 Conclusion

We gained some really good insights from this paper. We were able to test out 5 different variations of the model and compare the results achieved over a set of 5 different evaluation metrics. We had 3 broad set of model approaches that we worked on : a baseline model (fine - tuned model), Plug and Play Language model without fine-tuning, and a Plug and Play Language model with the fine-tuned model. After performing a quantitative analysis on these models and achieving the results, it can be concluded that our second approach (where we integrated the PPLM model with the fine tuned model) gave significantly better results, and outperformed the baseline model.

One of the challenges we faced was the unsatisfactory performance given by the Plug and Play model, due to which we had to introduce a second variation of the model (the second approach), which ended up giving us improved results when compared to the baseline model.

If we choose to continue working on the project in the future, there are further additional features we would add to the existing model that we proposed initially but weren't able to implement them. One of the things that we would add is the Persuasive/Non-Persuasive by utilising the CORPS Dataset and classify speeches based on if they are persuasive or not, based on the audience reaction labels. We would also add another

attribute to our model, which is the opinion of the speaker, that is whether the speaker is in support or not in support of the topic that they are speaking about. We would also like to evaluate the generated speeches on a more broader set of evaluation metrics in the future, which would give us much more accurate results. We would also introduce Prompt Tuning in our project in the future. We noticed that our prompts were a huge deciding factor in how the speech turned out. So, despite the attributes given to our model, we observed that speeches were heavily influenced by the prompts given. So, we can try and learn these prompts further with the help of prompt tuning.

References

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Celikyilmaz, A., Clark, E., and Gao, J. (2021). Evaluation of text generation: A survey.

Dathathri, S., Madotto, A., Lan, J., Hung, J., Frank, E., Molino, P., Yosinski, J., and Liu, R. (2020). Plug and play language models: A simple approach to controlled text generation.

Ficler, J. and Goldberg, Y. (2017). Controlling linguistic style aspects in neural language generation. *arXiv preprint arXiv:1707.02633*.

Gentzkow, Matthew, J. M. S. and Taddy, M. (2020). Congressional record for the 43rd-114th congresses: Parsed speeches and phrase counts.

Hatori, J., Kikuchi, Y., Kobayashi, S., Takahashi, K., Tsuboi, Y., Unno, Y., Ko, W., and Tan, J. (2018). Interactively picking real-world objects with unconstrained spoken language instructions. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3774–3781. IEEE.

Kassarnig, V. (2016). Political speech generation. *arXiv preprint arXiv:1601.03313*.

Nguyen, A., Clune, J., Bengio, Y., Dosovitskiy, A., and Yosinski, J. (2017). Plug play generative networks: Conditional iterative generation of images in latent space.

Niu, T. and Bansal, M. (2018). Polite dialogue generation without parallel data. *Transactions of the Association for Computational Linguistics*, 6:373–389.

Peng, N., Ghazvininejad, M., May, J., and Knight, K. (2018). Towards controllable story generation. In *Proceedings of the First Workshop on Storytelling*, pages 43–49.

Piwek, P. and van Deemter, K. (2003). Dialogue as discourse: Controlling global properties of scripted dialogue. In *Proceedings of the AAAI Spring Symposium on Natural Language Generation in Spoken and Written Dialogue*.

Tang, J., Zhao, T., Xiong, C., Liang, X., Xing, E. P., and Hu, Z. (2019). Target-guided open-domain conversation. *arXiv preprint arXiv:1905.11553*.

Ziegler, Z. M., Melas-Kyriazi, L., Gehrmann, S., and Rush, A. M. (2019). Encoder-agnostic adaptation for conditional language generation. *arXiv preprint arXiv:1908.06938*.