UKRAINIAN CATHOLIC UNIVERSITY

MASTER THESIS

Text generation with control conditions compliance

Author:
Oleksandra KONOPATSKA

Supervisor:
Andrii LIUBONKO

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in the

Department of Computer Sciences
Faculty of Applied Sciences

Lviv 2023

Declaration of Authorship

I, Oleksandra KONOPATSKA, declare that this thesis titled, "Text generation with control conditions compliance" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

"Progress in controllable text generation illuminates the harmonious interplay between human ingenuity and machine intelligence, pushing the boundaries of creative expression and amplifying our capacity to shape language with precision and purpose."

ChatGPT

UKRAINIAN CATHOLIC UNIVERSITY

Faculty of Applied Sciences

Master of Science

Text generation with control conditions compliance

by Oleksandra KONOPATSKA

Abstract

Controllable text generation has emerged as a significant research area, allowing the production of text with desired characteristics. In this work, we investigate the controllability of text generation, exploring the challenges of controlling various aspects of generated text, such as length, parts-of-speech (POS) structure, sentiment, and tense; in addition, we extend our analysis to the task of multi-conditional text generation, which entails the possibility of simultaneous control of several parameters of the generated text.

Our research is mainly based on fine-tuned GPT-2, an autoregressive transformer-based model. Using fine-tuned GPT-2, we managed to achieve notable progress in controlling the above-mentioned text attributes; we also present the results of experiments using other approaches, such as diffusion models and ChatGPT. The models are trained on our own dataset, meticulously curated in-house; the evaluation of the generation results is carried out using a comprehensive set of control, fluency, distinctiveness, and repetition metrics.

Through rigorous analysis, we assess the performance of studied models in terms of controllability. Length control, in particular, proved to be a challenging aspect, even when employing the largest available models. Nevertheless, our fine-tuned GPT-2 demonstrated promising results, showcasing its capabilities in generating text with desired characteristics.

Overall, our findings highlight the possibilities of controllable text generation using fine-tuned GPT-2 and other models. Our work contributes to the ongoing exploration of techniques for improving controllability in text generation. As this field continues to evolve, further research can build upon our analysis and methodologies to enhance controllability and pave the way for more sophisticated text generation systems.

Acknowledgements

I would like to thank for the opportunity to conduct this research: SQUAD company, who kindly provided a scholarship for studying at the Master's degree program in Data Science at the Ukrainian Catholic University; Andrii Liubonko, who has been an awesome supervisor and invaluable help in the process of this research; Ihor Markevych, who kindly gave advice and guided me on the right path; everyone who supported me in the process of this work; the Armed Forces of Ukraine for fighting for what is the most important in the world.

Contents

List of Figures

Dedicated to all those who guard the light

Chapter 1

Introduction and motivation

Generative modeling is one of the most quickly developing topics in the nowadays artificial intelligence world. Generative models show prominent results in different domains, such as image, audio, video, and text generation. Existing approaches to generation tasks, such as autoregressive models, flow-based models, Generative Adversarial Networks, energy-based models, and latent variable models, are capable of creating high-quality visual, auditory and textual samples [Tomczak, 2022].

The text generation domain is often in the spotlight nowadays, with openly available APIs and user-facing chatbots such as ChatGPT[1] and the ability to generate texts of different types and lengths up to whole novels. The most common approach for text generation is autoregressive models with huge Transformer-based language models [Brown et al., 2020]. These models show high quality of generation, and allow the range of opportunities to adapt the existing pre-trained models to specific tasks such as controllable text generation.

Another promising approach to text generation is diffusion models [Sohl-Dickstein et al., 2015, Li et al., 2022]. Early works on diffusion models produced competitive results on image [Ho, Jain, and Abbeel, 2020, Song et al., 2020] and audio [Chen et al., 2020, Kong et al., 2020] generation. Although applying diffusion models to text generation requires tuning the approach due to the discrete nature of the text data, the works in this field have already been done with promising results [Austin et al., 2021, Chen, Zhang, and Hinton, 2022, Strudel et al., 2022]. Moreover, recent works on text generation with diffusion models focus on the controllability and interpretability of the generation [Li et al., 2022, Han, Kumar, and Tsvetkov, 2022, Strudel et al., 2022].

While showing competitive text generation quality, both autoregressive models and diffusion models face some limitations, including long generation time, and limited generation control [Li et al., 2022, Han, Kumar, and Tsvetkov, 2022]; moreover, solving specific tasks, such as length control, can be quite tricky even for the largest models.

In our work, we investigate the possibilities of controllable text generation for different approaches. The main base of our research is GPT-2 model, fine-tuned for the tasks of length, POS (parts-of-speech), sentiment and tense control, as well as multi-conditional text generation, where requirements of control condition are set for parts-of-speech, sentiment and length at the same time. We also investigate the possibilities of ChatGPT to control length and parts-of-speech structure of the text generation, and provide some experiments with the existing approaches based on the idea of diffusion.

For this work, we created our own dataset, which is based on OpenWebText[2]; in order to use it for model training for conditional text generation tasks, we calculated researched control attributes on the dataset samples. We also constructed the set of

[1] https://openai.com/blog/chatgpt/

[2] https://huggingface.co/datasets/openwebtext

evaluation metrics, which allows us to assess different facets of the generation text, such as compliance with control condition, fluency, distinctiveness and repetition.

Fine-tuned GPT-2 model demonstrated plausible results for all the metrics in tasks of length, parts-of-speech, sentiment, tense control and multi-conditional generation. However, there is room for improvement, both in this method, and even in such a large public API as ChatGPT. This, as well as other limitations of the considered approaches, will be discussed later in the work in the relevant sections.

In summary, this work investigates the tasks and approaches to controllable text generation, with a particular emphasis on fine-tuning GPT-2 on the introduced dataset, specifically tailored to research objectives. Through empirical evaluation, we aim to analyze the possibilities of controllable text generation, assess the limitations of existing techniques, and propose potential avenues for future research.

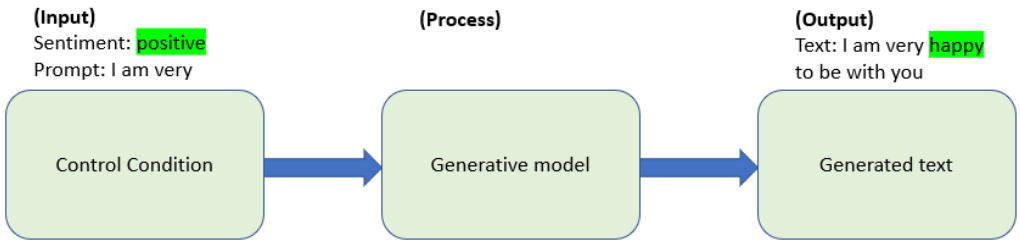
The next parts of the work have the following structure: chapter 2 is concentrated on the literature review – it provides an overview of the controllable text generation and existing approaches to this task; chapter 3 discusses approaches to the research: section 3.1 describes controllable generation tasks, section 3.2 speaks about models that we experimented with during our research, and section 3.3 relates to the dataset, constructed for the investigated tasks; chapter 4 demonstrates conducted experiments and their results – section 4.1 describes in detail metrics used for the evaluation of the generation results, and section 4.2 tells about specific experiments and their results; chapter 5 summarizes the work and provides conclusive remarks, as well as outlines possible future directions of the research.

Chapter 2

Related work

Controllable text generation is one of the key challenges of natural language generation. Its task is to generate a text corresponding to a certain control condition. Such a condition can be, in particular, sentiment, topic, syntactic structure, or inclusion of keywords. Zhang et al., 2022a propose a division of control conditions into semantic (sentiment, topic, lack of toxicity, etc.), structural (syntax tree, format, etc.), and lexical (inclusion of keywords, etc.). Zhang et al., 2022a also offer a generalized scheme of the controlled text generation system: control condition (input), generative model (process), and generated text (output) (Fig. 2.1).

FIGURE 2.1: Generalized scheme of the controllable text generation system based on "The IPO of controlled text generation" figure in Zhang et al., 2022a



Some earlier approaches to controllable text generation include sequential models and style embedding [Ficler and Goldberg, 2017, Li et al., 2016], Variational Autoencoders [Hu et al., 2018, Sohn, Lee, and Yan, 2015], Generative Adversarial Nets [Scialom et al., 2020, Wang and Wan, 2018], and Energy-based Models [Deng et al., 2020, Zhao, Mathieu, and LeCun, 2017]. These DL-based methods, however, heavily relies on large-scale datasets [Zhang et al., 2022a].

Today, the most widely used approach for text generation is large autoregressive language models [Brown et al., 2020, Chowdhery et al., 2022, Zhang et al., 2022b]. Fine-tuning such models with supervised data or even training them from scratch for controllable text generation is generally used to achieve the ability to control the generation result [Keskar et al., 2019]. At the same time, updating model parameters for each individual case is a rather large and expensive task, moreover autoregressive language models use a fixed order for a generation – left-to-right, which reduces the flexibility of the models for those tasks of controlled text generation that require the use of both left and right contexts.

One of the large autoregressive language models is GPT-2, developed by OpenAI in 2019 [Radford et al., 2019] – a sentence-generative language model that employs

only the decoder blocks from the Transformer architecture. Operating as a traditional language model, GPT-2 takes word vectors as an input and generates a probability distribution for the next word as an output. Its auto-regressive nature ensures that each token in a sentence is conditioned on the context of the preceding words. This autoregressive framework is built upon the Transformer's decoder component and incorporates masking mechanisms during training, allowing attention calculations to observe only the content before a given word and not the content after it. Fine-tuning such a pre-trained language model has shown promising results across various tasks [Zhang et al., 2022a].

On the other hand, non-autoregressive language models show good results on machine translation and speech-to-text tasks [Gu et al., 2017, Saharia et al., 2020]. However, these models are not well suited for language modeling [Ren et al., 2020].

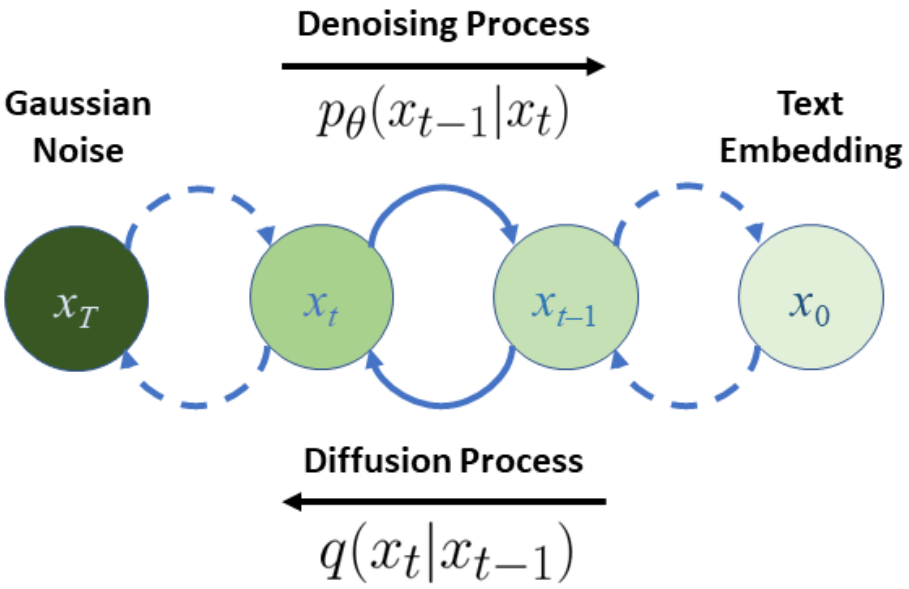
The limitations of autoregressive models have led to the appearance of light-weighted plug-and-play approaches [Dathathri et al., 2019, Yang and Klein, 2021, Krause et al., 2020, Liu et al., 2021], which consist in keeping the language model fixed and steering the generation process with external classifiers (potential functions). This way, plug-and-play approaches can both control satisfaction of the desired conditions (using a probabilistic potential function) and fluency of the generated text (using language model's probabilities). However, plug-and-play approaches inherit limitations of language models on which they are based (such as only left-to-right order of generation for autoregressive language models). Moreover, Li et al., 2022 shows that plug-and-play methods are only successful on attribute level (e.g., topic) control, and fail on more complex tasks (such as syntactic structure and semantic content).

Another approach to controllable text generation are diffusion models, which borrow the idea from non-equilibrium thermodynamics [Sohl-Dickstein et al., 2015]. The general idea of this approach is to "destroy" training data by gradually adding Gaussian noise; after that, the diffusion model is trained to reverse the diffusion process and recover the data. Then, the model can be used to generate the new data by successively applying the same learned "denoising" process to the randomly sampled noise. For several years, diffusion models have shown great performance in the continuous domain, providing state-of-the-art quality in the images [Ho, Jain, and Abbeel, 2020, Song et al., 2020, Nichol and Dhariwal, 2021, Kingma et al., 2021], audio [Chen et al., 2020, Kong et al., 2020], and video [Ho et al., 2022] generation tasks. The application of diffusion models to textual data has long been limited because of the need for additional adaptations due to the discrete nature of the data. Existing approaches to this adaptation mainly include extending diffusion models to discrete state spaces [Hoogeboom et al., 2021b, Austin et al., 2021, Han, Kumar, and Tsvetkov, 2022] and applying approximation and corruption processes on discrete data (embeddings, roundings, character, byte-level methods) [Hoogeboom et al., 2021a, Li et al., 2022, Gong et al., 2022, Chen, Zhang, and Hinton, 2022, Strudel et al., 2022].

The application of diffusion models to controllable text generation was introduced in 2022 with the Diffusion-LM model [Li et al., 2022]. Combining ideas of diffusion models and plug-and-play approaches, Diffusion-LM starts with Gaussian noise vectors and gradually performs denoising steps which produce a sequence of continuous latent representations up to vectors that correspond to words. Fig. 2.2 shows described diffusion and denoising processes for text embeddings generation.

While showing promising results on different controllable text generation tasks (semantic content, syntax tree, infilling etc.), Diffusion-LM demonstrates some limitations, namely: (1) higher perplexity compared to results of previous plug-and-play

FIGURE 2.2: Graphical model of diffusion and denoising processes for text embeddings generation



models; (2) significantly slow training and decoding processes. Other diffusion-based approached to conditional text generation include such models as SSD-LM [Han, Kumar, and Tsvetkov, 2022], SED [Strudel et al., 2022] and LD4LG [Lovelace et al., 2022].

Chapter 3

Approach

Our approach to solving the tasks of controllable generation is based on the following ideas: based on the classical approach to text generation (GPT-2), we aimed to check the possibility of its fine-tuning for various tasks of text generation, its limitations, pros and cons; we also wanted to test new emerging approaches and try to adapt existing diffusion solutions to the tasks of controllable generation; also, we wanted to conduct experiments on ChatGPT, particularly on those tasks that could potentially cause difficulties for it (length and parts-of-speech control). To implement and evaluate the results of the assigned tasks, we decided to create our own dataset and set of metrics.

The motivation and process of creating the dataset are detailed in section 3.3, and the description of the metrics is in section 4.1. Details and results of the conducted experiments are placed in section 4.2.

3.1 Controllable generation tasks

To assess the ability of the models to solve the problems of controllable text generation, the tasks of length, parts-of-speech, sentiment, and tense control were selected. We also added a multi-conditional generation task, which is designed to test the ability of models to simultaneously work with several control conditions. The idea is to evaluate the model's performance on various types of tasks: semantic (sentiment), structural (POS, length) and morphosyntactic (tense), as well as their combination.

3.1.1 Length control

In the task of length control we train the model to generate sentences with the given number of tokens, where number of tokens is the number of all words and punctuation symbols in the sentence. The parts of contractions are counted as the separate tokens: for example, contraction "I'm" consists of two tokens "I" and "'m".

3.1.2 Parts-of-speech control

In the task of parts-of-speech control we train the model to generate sentences according to the given sequences of parts-of-speech tags. POS sequences used to test the performance of the model are taken from the parts-of-speech structures of the real sentences of the validation dataset (see section 3.3 for dataset construction details).

3.1.3 Sentiment control

In the task of sentiment control we train the model to generate sentences with the given sentiment: positive, neutral or negative.

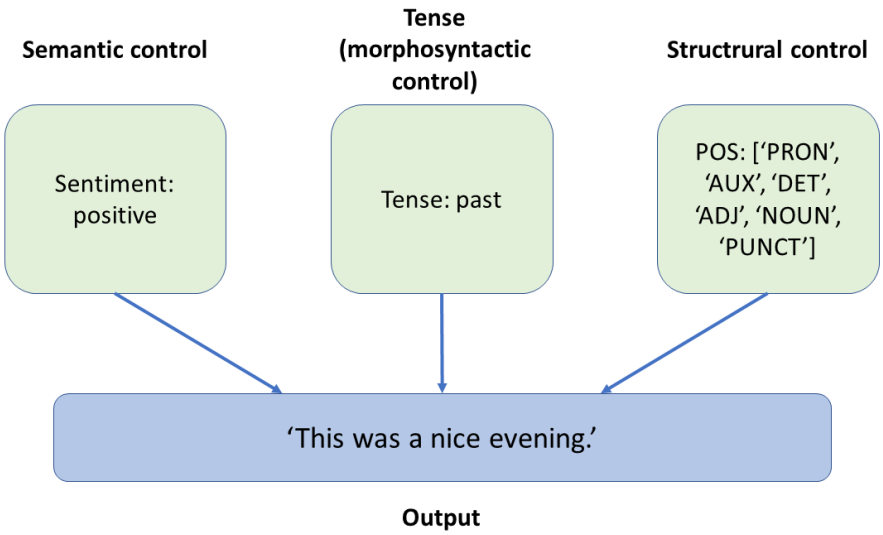
3.1.4 Tense control

In the task of tense control we train the model to generate sentences with the given tense. For this task, we used the simplified system of tenses division, and consider only "general present", "general future" and "general past" classes which include all the corresponding tenses.

3.1.5 Multi-conditional generation

In the task of multi-conditional generation we train the model to generate sentences which satisfies parts-of-speech, sentiment and tense control simultaneously. The example of multi-conditional generation is shown in Fig. 3.1.

FIGURE 3.1: Example of a multi-conditional text generation task with three control conditions



3.2 Models

3.2.1 GPT-2

For fine-tuning GPT-2 for our tasks of controllable text generation, we took pre-trained GPT-2 model available via HuggingFace[1]. Original GPT-2 model consists of 1.5 billion parameters and was trained on a WebText dataset of 8 million web pages.

GPT-2 was used in default set-up and architecture, and fine-tuned on our dataset (see section 3.3) for the tasks of length, parts-of-speech, sentiment, tense control, and multi-conditional text generation. As the result of the fine-tuning process, five models were obtained:

[1]https://huggingface.co/docs/transformers/model_doc/gpt2

- GPT-2 length
- GPT-2 pos
- GPT-2 sentiment
- GPT-2 tense
- GPT-2 pos_sentiment_tense

For fine-tuning each of the models learning rate 0.0001 and 10 maximum epochs were used; however, we set "patience" parameter to 3, which means that training stops 3 epochs after the loss doesn't improve. This way, training of all the models stopped after the 6th epoch.

Tokenizer of the GPT-2 was expanded with the following additional tokens:

```
ADDITIONAL_TOKENS = {
'WORDS': '<COND_LENGTH>',
'POS': '<COND_POS>',
'SENTIMENT': '<COND_SENT>',
'TENSE': '<COND_TENSE>',
'START': '<START>',
'END': '<END>',
'PAD': '<PAD>'
},
```

where *'WORDS'*, *'POS'*, *'SENTIMENT'* and *'TENSE'* are condition tokens, *'START'* and *'END'* indicates beginning and end of the sentence, and *'PAD'* is a placeholder token. The output layer of GPT-2 model has been changed accordingly to take into account the additional tokens.

An example of the sentence format with control conditions used when training a multi-conditional model:

```
<COND_LENGTH> 8 <COND_SENT> neutral <COND_TENSE> present
<START> This is a short article about sports.  <END>
```

An example of a prompt format with control conditions used to multi-conditional generation:

```
<COND_LENGTH> 8 <COND_SENT> neutral <COND_TENSE> present
<START>
```

An AWS g5.2xlarge instance with one GPU was used to train the model. Training of one model took about 12 hours on average.

3.2.2 Diffusion-based models

As part of this work, experiments with diffusion-based approaches were also conducted. We used Diffusion-LM[2] and minimal-text-diffusion[3] based on it, and tried to train them on our dataset.

A diffusion mode is a latent variable model. It models the data x_0 as a Markov chain $x_T...x_0$, where $x_T...x_1$ are latent variables with dimensionality equal to the dimensionality of x_0, and x_T is Gaussian [Li et al., 2022]. On each step of the chain, the Gaussian noise is gradually added to the data to get the approximate posteriors $q(x_t|x_{t-1})$ (Fig. 2.2).

[2]https://github.com/XiangLi1999/Diffusion-LM
[3]https://github.com/madaan/minimal-text-diffusion

The training process of the diffusion model is learned to reverse the diffusion process by training $p_\theta(x_{t-1}|x_t)$. The new data can then be generated with the same algorithm of denoising.

Diffusion-LM is controlled with the gradient-based method, which balances satisfaction of text fluency and required control conditions. minimal-text-diffusion is light-versioned adherent of Diffusion-LM model. It allows controllable text generation by combining the generative diffusion model with a classifier model.

An AWS g5.12xlarge instance with four GPUs was used to train Diffusion-LM, and AWS g5.2xlarge instance with four GPUs was used to train minimum-text-diffusion. Training of Diffusion-LM model took about 36 hours, and minimum-text-diffusion – about 3 hours.

3.2.3 ChatGPT

For conducting experiments with ChatGPT, the publicly available online version hosted on the OpenAI website[4] was used.

3.3 Dataset

For the tasks of our research, we decided to create our own dataset. The motivations for this are as follows:

- to have a full control over the training data, including size, attributes and other aspects of the dataset
- to be able to make a proper analysis of different facets of the generation tasks
- to our knowledge, there is no standardized benchmark for the problem of conditional text generation with multiple conditions in the literature. There are standard benchmarks for unconditional generation, paraphrasing, summarization, etc., but we are unaware of any well-established benchmark for our tasks. Accordingly, composition of own dataset, which meets the conditions necessary for research, appears to be a consistent solution.

3.3.1 Dataset preparation pipeline

To create our own dataset, we are using the following procedure:

1. We took as a basis the OpenWebText dataset[5], which is an open-source replication of the WebText dataset from OpenAI. WebText, in turn, is a corpus created from web pages collected according to the principle of quality.

2. We randomly sampled from OpenWebText the dataset of a 10000 paragraphs, each of which consists on average of 4.5 sentences; the resulted dataset consists of 451103 sentences from various-thematic web publications.

3. We performed the first cleaning of the dataset, which includes applying the following filters:

- whether a sentence has a string type
- whether a sentence has more letters than numbers, symbols and sum of numbers and symbols

[4] https://chat.openai.com
[5] https://huggingface.co/datasets/openwebtext

- whether a length of a sentence is between 4 and 80 words and punctuation inclusive
- whether a length of a sentence is more than 10 symbols
- whether a sentence doesn't contain @ symbol (email address) or "http" substring (link to the webpage)
- whether a sentence is completed and its last character is one of the following: !'".?»" Also, some sentences were not accurately separated and included punctuation (one symbol, such as opening quotation mark) from the next sentence; therefore, we also checked whether the third character from the end is one of the following: !'".?»", and if it is, deleted the last two characters and considered a sentence completed
- whether sentence contains less than or equal to 3 special symbols #$%&*+/<=>

After the first cleaning, the size of the dataset decreased to 378852 sentences.

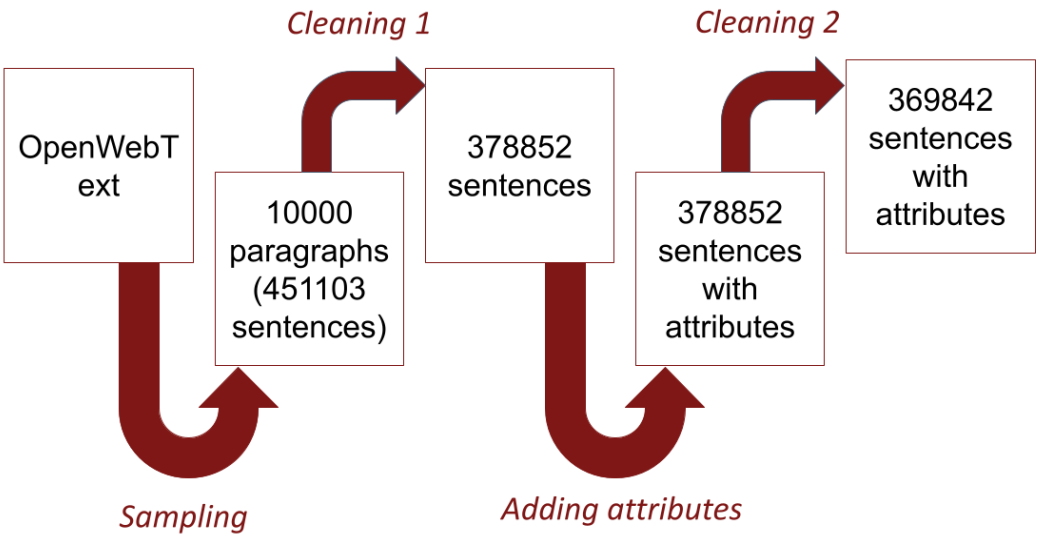
4. We calculated or constructed using available models the set of attributes for each sentence. The detailed description of the attributes is presented in the subsection 3.3.2.

5. We performed the second cleaning of the dataset, which was made based on the mismatches and errors in the generated attributes, such as mismatch between the sentence length, the number of parts-of-speech tags, and the value of the sentence length attribute.

After the second cleaning, the size of the dataset is 369842 sentences.

Fig. 3.2 shows schematic image of the dataset preparation pipeline.

FIGURE 3.2: Dataset preparation pipeline



3.3.2 Attributes

The set of attributes was calculated or constructed using available models for each sentence in the dataset. This set of attributes include:

1. Number of symbols in the sentence ("Number of symbols").

This attribute was used only for purposes of cleaning and visualization of the distribution of sentence lengths (see subsection 3.3.4).

2. Number of words in the sentence ("Number of words").

This attribute corresponds to the number of words and punctuation symbols in a sentence. For example, length of the sentence *She loves her two dogs.* is 6.

3. Parts-of-speech tagging ("Parts-of-speech"). The list of UPOS (universal part-of-speech) tags of all words and punctuation symbols in a sentence, constructed using spaCy tools[6].

4. Sentiment ("Sentiment analysis"). Sentiment of a sentence, constructed using twitter-roberta-base-sentiment model from Cardiff NLP [Barbieri et al., 2020]. The values of the attribute are as follows: 0 – negative, 1 – neutral, 2 – positive. The output of the model is a dictionary of sentiment distribution between all three values (which sums up to 1), but for the convenience of training we use for this attribute only the index of the maximum value.

To select a model for constructing this attribute, four approaches were compared: Polarity of the TextBlob by spaCy[7], eng_spacysentiment by spaCy[8], sentiment by Stanza[9], and twitter-roberta-base-sentiment model from Cardiff NLP[10]. Four approaches were compared on the test dataset of 60 sentences (20 with positive sentiment, 20 with neutral sentiment and 20 with negative sentiment), constructed by ChatGPT from the literature quotes. Sentiment tags, as well as performance of the approaches were evaluated manually; twitter-roberta-base-sentiment model showed significantly better results than other approaches.

5. Tense ("Tense").

Tense attribute was constructed based on the scheme of interpreting tags proposed in this Stack Overflow answer[11]. For the convenience of training we used only three tense values: future, present and past.

Some, especially long, sentences are composed of several parts with different tenses. In such cases, the attribute returned a set with all the tenses present in the sentence. Later, such sentences were filtered out and not used for training (see subsection 3.3.3).

3.3.3 Training dataset

For training the models, the smaller version of the dataset were sampled. The reasons behind this decision are:

1. To speed-up training process. Both GPT-2 and diffusion-based approaches require large computational resources, which complicates the iterative process of research when training on a large dataset. With further research, the final models can be trained on a larger dataset, which is expected to improve the solving of controllable generation tasks and increase the vocabulary.

2. To balance training dataset on sentiment and tense attributes. As will be shown in the section 3.3.4, the large dataset have high prevalence of sentences with

[6] https://spacy.io/usage/linguistic-features
[7] https://spacy.io/universe/project/spacy-textblob
[8] https://spacy.io/universe/project/eng_spacysentiment
[9] https://stanfordnlp.github.io/stanza/sentiment.html
[10] https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment
[11] https://stackoverflow.com/a/70976698

neutral sentiment, and lack of sentences with future tense. To ensure the stable work of the models, we balanced the labels of these attributes in sampled smaller dataset.

3. To ensure that all the sentences in training dataset have only one detected tense. For some sentences from a large dataset, the detected tense is either empty or contains more than one tense. Such samples can make it difficult to train models on the tense attribute, so it was important that the training dataset consisted only of those sentences for which the tense was uniquely determined.

The size of the smaller (*training*) dataset is 100000 sentences.

Also, validation dataset with 10000 sentences were sampled. It was used to assess loss of the model after each epoch of fine-tuning GPT-2.

3.3.4 Datasets description

After the second cleaning, the size of the larger (*general*) dataset is 369842 sentences; the size of the training dataset is 100000 sentences. Each of the datasets consists of six columns:

- 'text' feature, which contains the sentences of the dataset, one sentence per row;

and features, which represent attributes of the corresponding sentences:

- 'Number of symbols';
- 'Number of words';
- 'Parts of speech';
- 'Sentiment analysis';
- 'Tense'

Distributions of attributes 'Number of symbols' and 'Number of words' are presented on Fig. 3.3 and Fig. 3.4; distribution of 'Sentiment analysis' feature is presented on Fig. 3.5 and Fig. 3.6; distribution of 'Tense' feature is presented on Fig. 3.7 (for the general dataset, only those sentences in which 'Tense' attribute is defined and contains only one value were taken into account) and Fig. 3.8.

The figures show an imbalance of classes for sentiment and tense attributes in the general dataset, and their balance in the training dataset. In addition, the graphs demonstrate that the distribution of sentence lengths is the same for both datasets.

FIGURE 3.3: Distributions of 'Number of symbols' and 'Number of words' attributes in the general dataset

	Number of symbols	Number of words
count	369842.000000	369842.000000
mean	119.581405	23.086572
std	70.025735	12.714929
min	9.000000	3.000000
25%	67.000000	13.000000
50%	108.000000	21.000000
75%	159.000000	30.000000
max	515.000000	80.000000

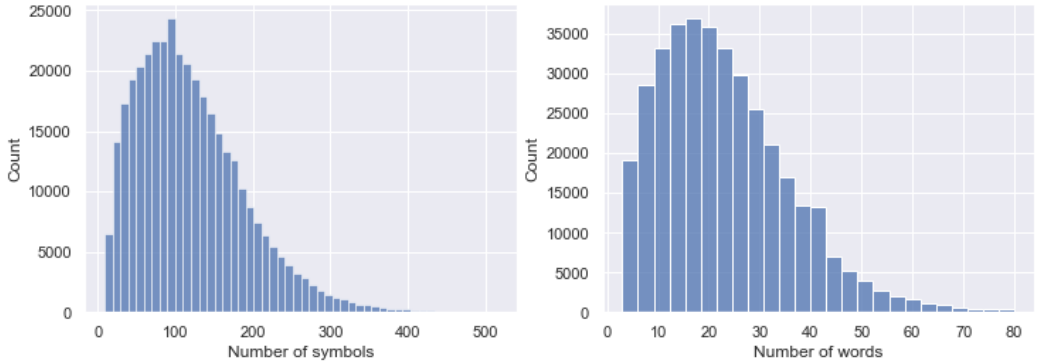


FIGURE 3.4: Distributions of 'Number of symbols' and 'Number of words' attributes in the training dataset

	Number of symbols	Number of words
count	100000.000000	100000.000000
mean	115.140350	22.168130
std	63.570194	11.422009
min	9.000000	3.000000
25%	68.000000	14.000000
50%	105.000000	20.000000
75%	151.000000	29.000000
max	473.000000	80.000000

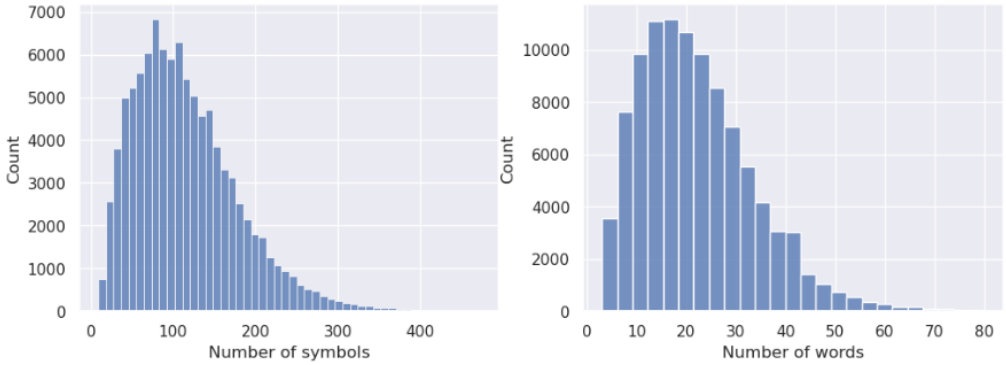


FIGURE 3.5: Distribution of 'Sentiment analysis' attribute in the general dataset

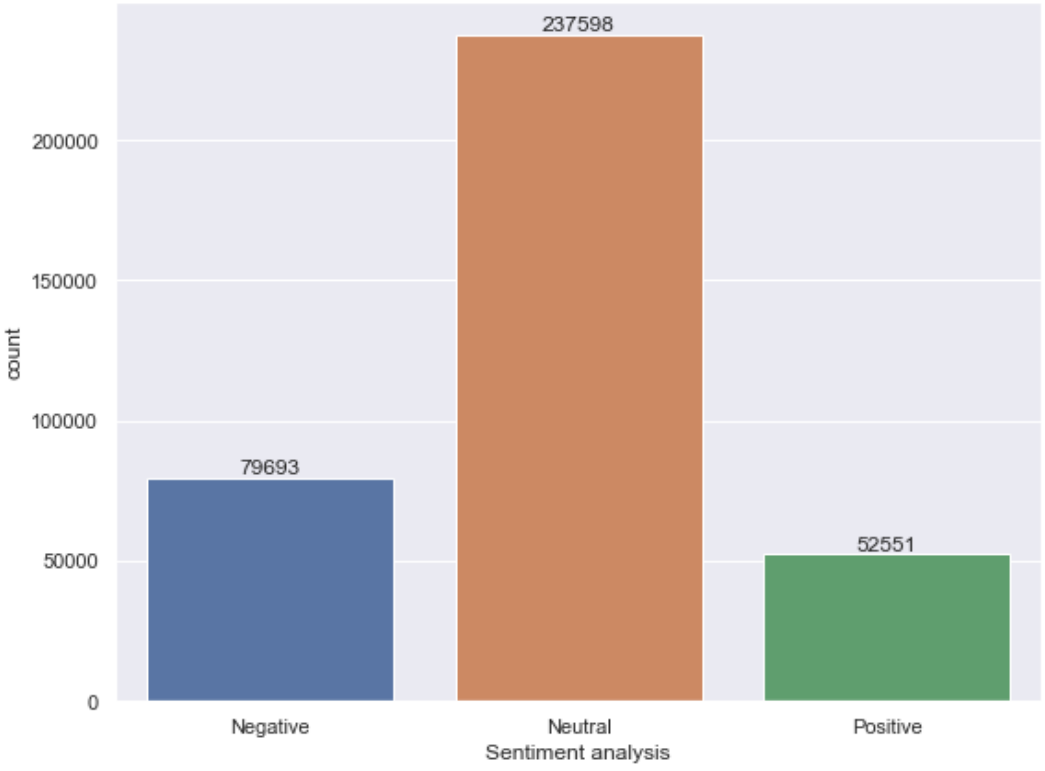


FIGURE 3.6: Distribution of 'Sentiment analysis' attribute in the training dataset

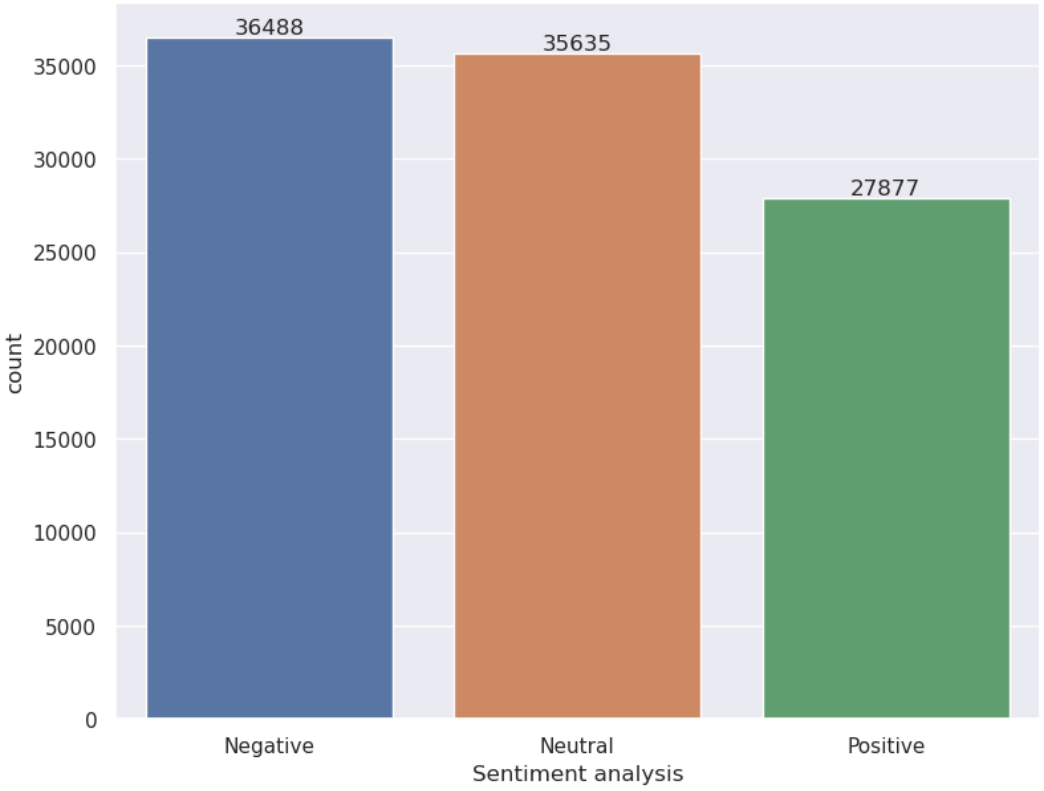


FIGURE 3.7: Distribution of 'Tense' attribute (for those sentences, in which 'Tense' attribute is defined and contains only one value) in general dataset

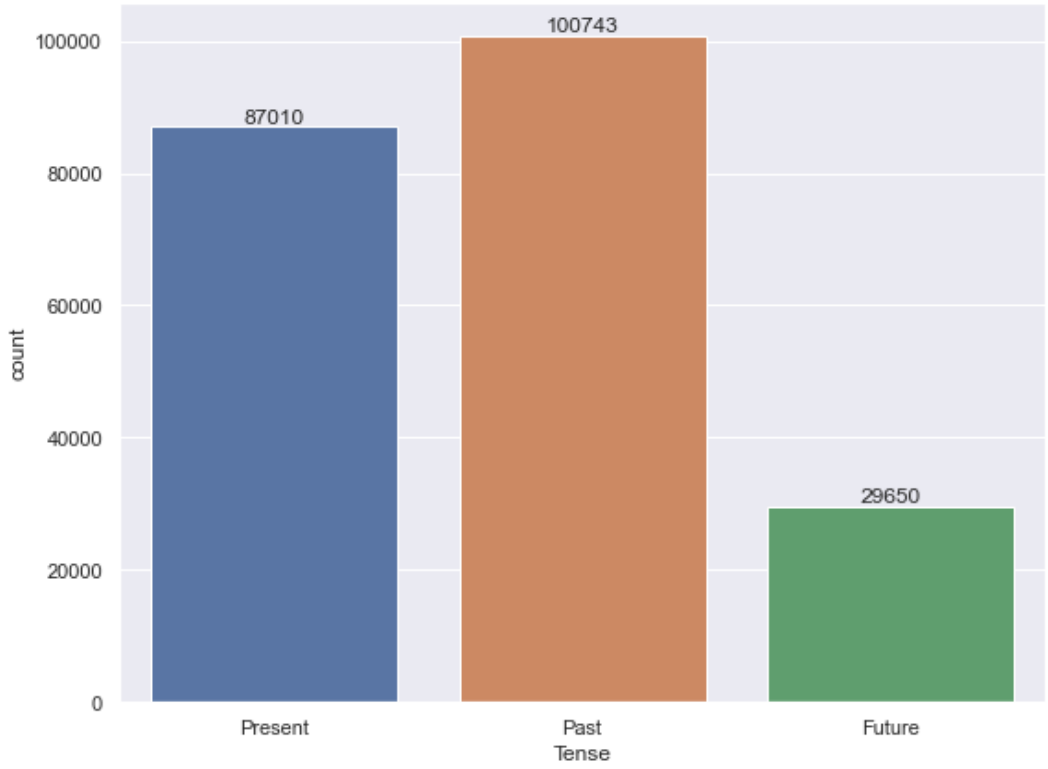
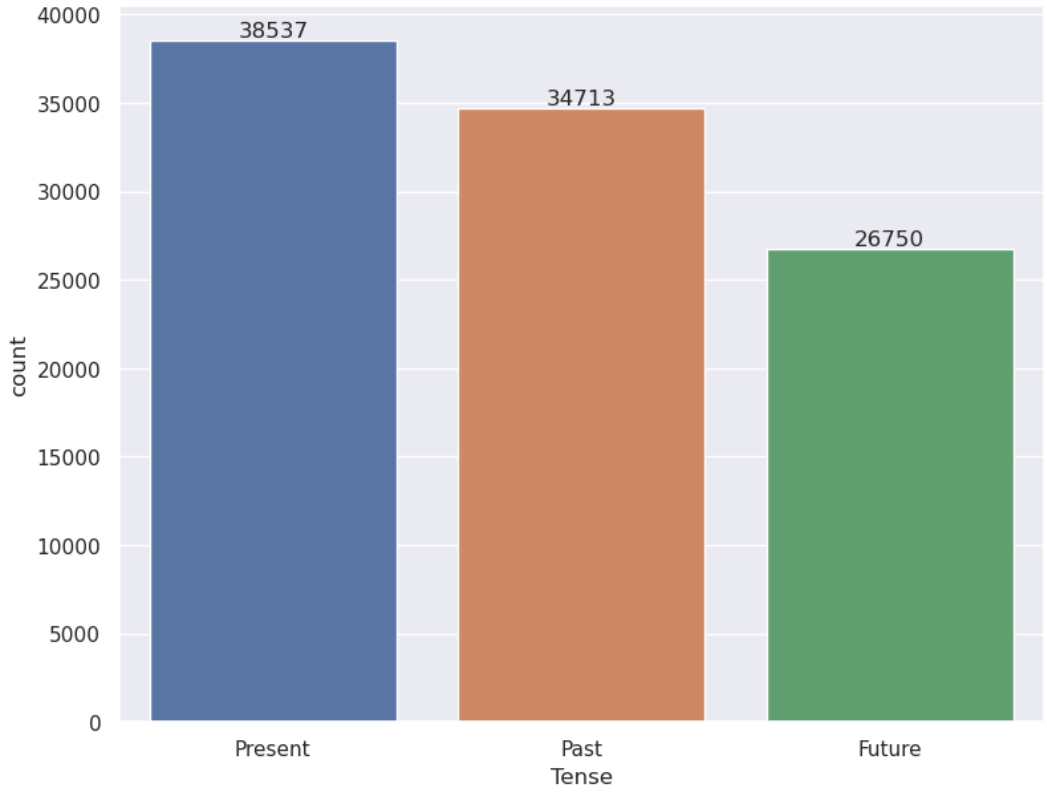


FIGURE 3.8: Distribution of 'Tense' attribute in training dataset



Chapter 4

Experiments and evaluation

4.1 Metrics

In the tasks of controllable text generation, it is necessary to consider several quality aspects of the generated text. In particular, it is important not only to fulfill the given control condition but also to produce qualitative (fluent) and diverse samples. This is crucial in order to avoid the generation of "unreadable" or identical texts; also, it's important to ensure that generated text doesn't directly reproduce the training samples.

To ensure the quality of all four aspects, we are using the evaluation methodology based by evaluation scheme proposed by the authors of the Diffusion-LM [Li et al., 2022] and SSD-LM [Han, Kumar, and Tsvetkov, 2022] models, but additionally processed and adjusted to the specific control conditions of our work:

1. **Control metrics.** To evaluate compliance with the control condition, an accuracy metric of exact match to the condition is used; for the tasks of length and sentiment control, accuracy metric of approximate match to the condition is also used; we also use the mean deviation metric to assess the length difference; task of POS control has several additional control compliance metrics, which will be discussed in subsection 4.1.4.

2. **Fluency metrics.** The fluency of the model is assessed using the perplexity metric.

3. **Distinctiveness metrics.** Distinctiveness metrics are used to assess diversity. They include such metrics as the average percentage of distinct n-grams in the output samples and the percentage of output samples that begin or end with identical phrases.

4. **Repetition metrics.** Repetition metrics are used to ensure that generated samples differ from the training data. We use average percentage of unseen n-grams to calculate these metrics.

4.1.1 Length control metrics

For the tasks with the length control, three metrics are used to assess the compliance with control condition:

1. Exact length: percentage of sentences with the exact match of the specified control length with the number of words and punctuation symbols in the generated sentence.

2. Approximate length: percentage of sentences with the approximate (+- 3 words or punctuation symbols) match of the specified control length with the number of words and punctuation in the generated sentence. The approximate length metric aims to examine in more detail how well the model is able to capture the general trend in length in cases where the length condition is a large number and an

exact match with the control condition becomes significantly more complex task for the model.

3. Mean deviation of length: the mean value of the deviations of the lengths of the generated sentences from the specified control lengths conditions:

$$\underset{i\in[1,n]}{mean}(|(\text{length of generated sentence i}) - (\text{control length condition i})|)$$

4.1.2 Sentiment control metrics

For the tasks with the sentiment control, two metrics are used to assess the compliance with control condition:

1. Exact sentiment: percentage of sentences with the exact match of the specified control sentiment with the defined sentiment of the generated sentence. The definition of the sentiment of the generated sentences is made using the same twitter-roberta-base-sentiment model that was used to calculate the sentiment attribute in the training dataset.

2. Approximate sentiment: percentage of sentences with the approximate (+- 1 "sentiment level") match of the specified control sentiment with the defined sentiment of the generated sentence. The approximate sentiment metric aims to "smooth" the boundaries between "sentiment levels", which in certain cases can be quite close – negative (0) and neutral (1); neutral (1) and positive (2).

4.1.3 Tense control metrics

For the tasks with the tense control, one metric is used to assess the compliance with control condition:

1. Exact tense: percentage of sentences with the exact match of the specified control tense with the defined tense of the generated sentence. The definition of the tense of the generated sentences is made using the same approach which is used to calculate the tense attribute in the training dataset (subsection 3.3.2). Only those sentences in which 'Tense' attribute is defined and contains only one value are used to calculate this metric.

4.1.4 Parts-of-speech control metrics

For the tasks with the parts-of-speech control, four metrics are used to assess the compliance with control condition:

1. Exact POS: percentage of sentences with the exact match of the specified control sequence of parts-of-speech tags with the defined parts-of-speech tagging of the generated sentence. The part-of-speech tagging of the generated sentences is made using the same spaCy tools, used to construct Parts-of-speech attribute in the training dataset (subsection 3.3.2).

2. Presence of unique POS in control sequence: mean value of the percentages of unique POS tags, presented both in the control sequence of tags and sequence of POS tags of the generated sentence:

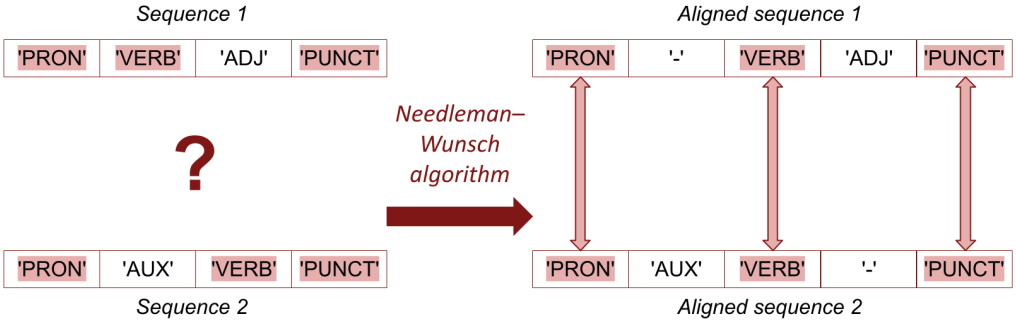
$$\underset{i\in[1,n]}{mean}\frac{\{[\text{set of POS tags in control i}] \cap [\text{set of POS tags in generated sentence i}]\}}{\{[\text{set of POS tags in control i}]\}}$$

3. POS n-grams match: mean value of the percentages of n-grams of POS tags, presented both in the control sequence of tags and sequence of POS tags of the generated sentence. In evaluating of the performance of the models we used POS bigrams match and POS trigrams match metrics.

$$\underset{i\in[1,n]}{mean}\frac{\{[\text{n-grams of POS in control i}]\cap[\text{n-grams of POS in generated sentence i}]\}}{\{[\text{n-grams of POS in control i}]\}}$$

4. Alignment POS: the idea of this metric comes from the assumption that POS tags in the generated sentence can differ from control sequence with some gaps or extra tags. In this case, direct evaluation of the exact POS match is too strict, while presence of unique POS in control sequence or POS n-grams match are too vague metrics. Therefore, there is a need to align both sequences, and to calculate the percentage of matched tags only after that. To align the sequences we use the Needleman–Wunsch algorithm – the algorithm presented in 1970 by Needleman and Wunsch, 1970 for bioinformatic uses of aligning protein or nucleotide sequences. After obtaining aligned sequences, we calculate mean value of the percentages of the matched tags in all the generated sentence. The example of the POS sequences alignment is shown in the Fig. 4.1.

FIGURE 4.1: Example of the POS sequences alignment with the Needleman-Wunsch algorithm



4.1.5 Fluency metrics

For ensuring the fluency of the generated text, we are using one metric:

1. Perplexity: we calculate the mean perplexity of the generated samples using HugginFace Perplexity model[1] [Jelinek et al., 2005]. The lower perplexity means the higher fluency of the model.

4.1.6 Distinctiveness metrics for all models

For assessing the distinctiveness of the generated text (in other words, checking if model doesn't generate same or very similar sentences) we use four metrics:

[1]https://huggingface.co/spaces/evaluate-metric/perplexity

1. Distinct n-grams in a sentence: we calculate the mean of sentence-level percentages of the unique n-grams. With this metric we evaluate whether the model use repetitive words while generating a sentence. In evaluating of the performance of the models we used distinct unigrams, bigrams and trigrams.

2. Distinct n-grams in all sentences: we calculate the percentage of unique n-grams in all the generated sentences. With this metric we evaluate how distinct are n-grams in all the generation result. In evaluating of the performance of the models we used distinct unigrams, bigrams and trigrams.

3. Distinct endings: we calculate the percentage of unique endings in all the generated sentences. Identical endings of the generated sentences is an encountered problem of the not enough trained generative models; with this metric we would like to ensure that generated sentences have distinct endings. In the evaluating of the performance of the models we calculate this metrics for 2 and 3 ending tokens.

4. Distinct beginnings: we calculate the percentage of unique beginnings in all the generated sentences. Identical beginnings of the generated sentences is an encountered problem of the not enough trained generative models; with this metric we would like to ensure that generated sentences have distinct beginnings. In the evaluating of the performance of the models we calculate this metrics for 2 and 3 beginning tokens.

4.1.7 Distinctiveness metrics for models with POS control

These metrics are aimed to ensure that generation results are different for the same POS condition sequences. For ensuring distinctiveness of generation under the equivalent POS tags, we generate 5 sentences for each POS control sequence, and calculate three following metrics:

1. Distinct n-grams for repeated POS: we calculate the mean of 5-sentence-level percentages of the unique n-grams (in other words, we find the percentage of the unique n-grams in every 5 sentences, and then calculate the mean of all the percentages). In evaluating of the performance of the models we used distinct bigrams, trigrams and five-grams.

2. Distinct endings for repeated POS: we calculate the mean of 5-sentence-level percentages of the unique endings (in other words, we find the percentage of the unique endings in every 5 sentences, and then calculate the mean of all the percentages). In evaluating of the performance of the models we calculate this metrics for 2 and 3 ending tokens.

3. Distinct beginnings for repeated POS: we calculate the mean of 5-sentence-level percentages of the unique beginnings (in other words, we find the percentage of the unique beginnings in every 5 sentences, and then calculate the mean of all the percentages). In evaluating of the performance of the models we calculate this metrics for 2 and 3 beginning tokens.

4.1.8 Repetition metrics

These metrics are aimed to ensure that generation results don't repeat training data. We use one metric for assessing uniqueness of the generated sentences:

1. Repetition n-grams: we calculate the percentage of unique n-grams in the generated sentences (in other words, such n-grams which are presented in generation results but not presented in the training data). In evaluating of the performance of the models we used distinct trigrams and five-grams.

4.2 Experiments and results

4.2.1 Experiments with GPT-2

Experiments with GPT-2 were carried out using the models described in subsection 3.2.1. Each of the fine-tuned models was given an input of 1000 samples containing the appropriate control conditions. As a result, 1000 generated sentences were obtained for each of the models, which were evaluated using the metrics discussed in section 4.1. Additional generation was carried out for the *GPT-2 length* model to test the dependence of the metric results on the number of generated sentences. Also, to evaluate the *GPT-2 pos* model, two different sets of sentences were generated: without repetitions of POS sequences and with repetitions (5 sentences per one control sequence).

GPT-2 length

To evaluate this model, a set of length tokens was created, the distribution of which repeats the distribution of sentence lengths in the training dataset (Fig. 3.4).

The model demonstrated an exact match with the control condition in 42.3% of the generated sentences, an approximate match with the control condition in 93.1% of the generated sentences, and mean deviation of 1.551. The model demonstrated the best results in matching with the control condition when generating short sentences: for sentences of up to 10 words, the exact match with the control condition is 65.1%, approximate match with the control condition is 100%, and mean deviation is 0.37. As sentences length increases, the percentage of exact matches with the control condition decreases, but for sentences up to 30 words, the approximate match is still 100%, and slightly decreases to 92.6% for sentences between 30 and 40 words. For sentences longer than 40 words, the exact and approximate match with the control condition drops sharply, while mean deviation grows, and after 50 words the model can no longer generate sentences with the given number of words (mean deviation more than 16 words). The values of the control metrics depending on the desired length of the generated sentences are presented in Fig. 4.2 and Fig. 4.8.

The perplexity of the model is 164.7, and the generation results also showed good results on the distinctiveness and repetition metrics (see Appendix B for a table with the values of all metrics).

We conducted an additional experiment to evaluate the dependence of the metric values on the number of generated samples; for this, we generated 2000 sentences with given length control conditions. The distribution of the values of the length tokens provided for the model input prompts also repeats the distribution of the sentence lengths of the training dataset. On 2000 sentences, the model showed similar results in compliance with the condition of control, perplexity, and repetition, but, as expected, showed slightly worse results on almost all distinctiveness metrics.

GPT-2 pos

To evaluate this model, we gave the model prompts with sequences of parts-of-speech tokens obtained from the structure of real sentences. For both cases (without repetitions and with repetitions), the model demonstrated good results of match with the control condition: 41.2% and 39.3% for exact match with a given POS sequence and 92.1% and 92.4% for alignment POS (results on other control metrics are presented in Appendix B).

The *GPT-2 pos* model showed worse perplexity than other models: 324 and 313.8 for generation without repetitions and with repetitions. This is most likely due to the specific sequences of tags given as input for generation. Although they are

FIGURE 4.2: Length control metrics depending on the values of the control condition for 1000 and 2000 generated samples of *GPT-2 length* model

Model	Length	Number of sentences	Control metrics		
			Exact length	Appr length	Mean length deviation
GPT-2 length 1000	<=10	149	65.1%	100.0%	0.37
	11-20	374	47.9%	100.0%	0.6
	21-30	273	39.6%	100.0%	0.77
	31-40	136	28.7%	92.6%	1.29
	41-50	43	0.0%	20.9%	7.72
	51-60	15	0.0%	0.0%	16.93
	61-70	7	0.0%	0.0%	27.43
	71-80	3	0.0%	0.0%	36
GPT-2 length 2000	<=10	283	71.7%	100.0%	0.3
	11-20	823	56.2%	99.6%	0.52
	21-30	632	40.1%	99.8%	0.74
	31-40	324	27.2%	93.8%	1.29
	41-50	101	0.0%	16.3%	7.51
	51-60	31	0.0%	0.0%	16.42
	61-70	10	0.0%	0.0%	27.6
	71-80	0	0.0%	0.0%	37.67

taken from real sentences, they often contain quite complex combinations of parts of speech and punctuation symbols, which affects the fluency of the generated sentences.

The models showed good results of distinctiveness metrics, and the best values of repetition metrics among all models, which is probably also related to the uniqueness of the given tag sequences.

Testing the model on repeated tag sequences showed excellent results: for the same prompts with parts-of-speech tag sequences, the model generates different results.

GPT-2 sentiment

To evaluate this model, we used an even distribution of control conditions corresponding to different values of sentiment – positive, neutral and negative.

The model performed well: 73.6% for an exact match with the control condition and 99.1% for an approximate match with the control condition. The model showed the best results for neutral sentiment (83.2% exact match), and good results for positive and negative sentiment (69.1% and 68.5% exact match, respectively). The values of the control metrics depending on the desired sentiment of the generated sentences are presented in Fig. 4.3.

Perplexity of the model is the best among all considered GPT-2 models – 137. Distinctiveness and repetition metrics are similar to the results of other models.

GPT-2 tense

To evaluate this model, we used an even distribution of control conditions corresponding to different values of tenses – past, present and future.

The model showed the best results for control conditions among all models – 98.7% exact match with the control condition. It is worth noting that this metric

FIGURE 4.3: Sentiment control metrics depending on the values of the control condition for *GPT-2 sentiment* and *GPT-2 pos_sentiment_tense* models

Model	Sentiment	Number of sentences	Control metrics	
			Exact sentiment	Appr sentiment
GPT-2 sentiment	0 (negative)	333	68.5%	98.8%
	1 (neutral)	334	83.2%	100.0%
	2 (positive)	333	69.1%	98.5%
GPT-2 POS sentiment tense	0 (negative)	333	64.9%	98.5%
	1 (neutral)	334	75.1%	100.0%
	2 (positive)	333	47.1%	94.6%

was calculated among sentences for which it was possible to unambiguously determine the tense on the post-generation check (956 sentences out of 1000). The model demonstrated the best results in the generation of sentences in the future tense (99.4%); for the past and the present tenses, the indicators of the match with the control condition turned out to be the same – 98.4%. The values of the control metrics depending on the desired tense of the generated sentences are presented in Fig. 4.4.

FIGURE 4.4: Tense control metrics depending on the values of the control condition for *GPT-2 tense* and *GPT-2 pos_sentiment_tense models*

Model	Tense	Number of sentences	Control metrics
			Exact tense
GPT-2 tense	past	315	98.4%
	present	322	98.4%
	future	319	99.4%
GPT-2 POS sentiment tense	past	320	96.9%
	present	279	98.6%
	future	235	79.1%

Perplexity of the model is 144.5. Distinctiveness and repetition metrics are similar to the results of other models.

GPT-2 pos_sentiment_tense

The input of this model is fed simultaneously with three control conditions. For the values of the control condition, we used the same approaches as for the single-conditional models: sequences of parts-of-speech tags from real sentences and a even distribution of sentiment and tense values.

The model performed slightly worse in terms of control than almost all single-conditional models: 11.2% worse for exact sentiment, 1.4% worse for approximate sentiment, 6.3% worse for exact tense (the value of the exact match with the tense

control condition was also calculated among sentences for which it was possible to unambiguously determine the tense on the post-generation check). However, the values for compliance with the POS control conditions almost didn't change. The multi-conditional model showed different results than the single-conditional models on the distribution of control metric values depending on the control condition values: the worst results for positive sentiment (47.1% exact match against 64.9% for negative sentiment and 75.1% for neutral sentiment) and future tense (79.1% exact match against 96.9% for the past tense and 98.6% for the present tense). The values of the control metrics depending on the desired sentiment and tense of the generated sentences are presented in Fig. 4.3 and 4.4.

Perplexity of the multi-conditional model is the highest among all models – 354.1; on the other hand, distinctiveness and repetition metrics showed the best results. All these facts are obviously related to increasing the complexity of the generated sentences due to combination of control conditions.

4.2.2 Experiments with diffusion-based models

Despite high expectations for the performance of diffusion models, based on successful examples of their performance in generation tasks, including recent works of controllable text generation, we were unable to achieve decent results when training Diffusion-LM and minimal-text-diffusion on our data (an example of one of the generated samples using minimal-text-diffusion is shown in Fig. 4.5).

FIGURE 4.5: Example of one of the generated samples using minimal-text-diffusion trained on our dataset

```
==================== DDPM Denoising Step = 1800 | Sample #10 ====================
[step 1800] the post far heaven found " to her, only well to one 18 to -. be...
mouth in new running whether an ga average regardless as hanageal across stories
the 150 your night gas the littlen views cast glass brilliant dark in
```

Most likely, the difference between the results of the models on the datasets presented in the original examples and our dataset lies in the difference in the datasets complexities. The authors of Diffusion-LM use two datasets to train the model: E2E, which is a collection of restaurant reviews, and ROCStories, a dataset of five-sentence stories. These datasets are quite subject-specific and have a rather small and simple vocabulary, which allows the models to work in certain narrow domains with limited number of words and show good generation results. minimal-text-diffusion, as a smaller model, uses even simpler training datasets in the published examples.

With the available configurations, none of these models were able to learn on the dataset we created, and the long time and large resource consumption of Diffusion-LM training imposes limitations on the possibilities of experiments.

We expect that for future work, one of the following approaches may give better results:

1. More training time/larger training dataset/more steps in the diffusion and denoising processes, which, accordingly, requires more computing power.

2. And/or development of own diffusion approaches or changing the architecture of existing solutions for our tasks.

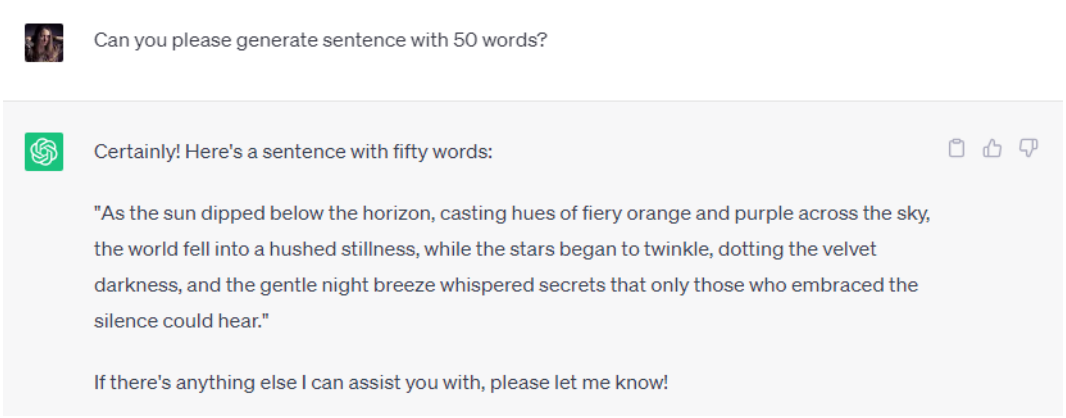
4.2.3 Experiments with ChatGPT

The task of generating text with a given sentiment and tense is apparently straightforward for ChatGPT. We decided to test its capabilities in generating sentences with a given length and parts-of-speech structure.

ChatGPT length

ChatGPT could not "understand" the task of generating text with a given number of words and punctuation symbols, so we concentrated on the task of generating text with a given number of words only. The metrics of compliance with the control conditions remained similar – the exact number of words, the approximate (+- 3) number of words and mean deviation of length. An example of prompt for generating sentences with a given number of words is shown in Fig. 4.6.

FIGURE 4.6: Example of prompt for generating sentences with ChatGPT with a given number of words



We generated 146 sentences with different length conditions. By exact and approximate match of the control condition, ChatGPT performed worse than GPT-2: 28.8% exact match with the control condition and 87% approximate match with the control condition. However, it showed similar value of mean deviation – 1.53, and by examining distributions of metrics depending on control condition we see that ChatGPT performance is much more stable across the different desired lengths. Unlike *GPT-2 length* model, ChatGPT is capable of generating longer texts with desired length. Weaker performance of GPT-2 on longer sentences is likely due to the lack of such sentences in the training data, which prevented the model from learning to generate long sentences of a given length. In contrast, ChatGPT has no such restrictions. However, it is also important to note that starting with 60 words, ChatGPT generates not a single sentence, but a text of a given length.

The values of the control metrics depending on the desired length of the generated sentences are presented in Fig. 4.7. The comparisons between the performances of GPT-2 length and ChatGPT models, which demonstrate the more stable performance of ChatGPT, are shown in Fig. 4.8.

ChatGPT, as expected, showed the best value of perplexity – 78.1. However, according to the distinctiveness metrics, *GPT-2 length* model turned out to be better in some of the metrics – in particular, ChatGPT often generates sentences that start the same way.

ChatGPT POS

In order to test the ability of ChatGPT to generate sentences based on the given parts-of-speech structure, we generated 50 sentences with 10 control sentences, which

FIGURE 4.7: Length control metrics depending on the values of the control condition for ChatGPT

Model	Length	Number of sentences	Control metrics		
			Exact length	Appr length	Mean length deviation
ChatGPT length	<=10	30	43.3%	100.0%	0.73
	11-20	30	26.7%	96.7%	1.07
	21-30	30	33.3%	93.3%	1.4
	31-40	30	16.7%	73.3%	2.2
	41-50	15	20.0%	66.7%	2.53
	51-60	5	20.0%	80.0%	2
	61-70	3	33.3%	100.0%	1
	71-80	3	33.3%	33.3%	3.67

FIGURE 4.8: Comparisons between the exact match and mean deviation metrics of *GPT-2 length* model for 2000 sentences and ChatGPT

were previously used for experiments with *GPT-2 POS* model. An example of prompt for generating sentences based on given parts-of-speech structure is shown in Fig. 4.9.

ChatGPT could not generate sentences that fully correspond to the given structure, but generated approximate variants: exact match with the given POS is 0%, but alignment POS value is 49.2%. Most likely, this is due to the fact that ChatGPT was not explicitly trained on these parts-of-speech tags, and has difficulties with their recognition.

In addition, attempts to generate sentences according to the given structure significantly worsened the perplexity of the generation results, which now amounts to 537.96.

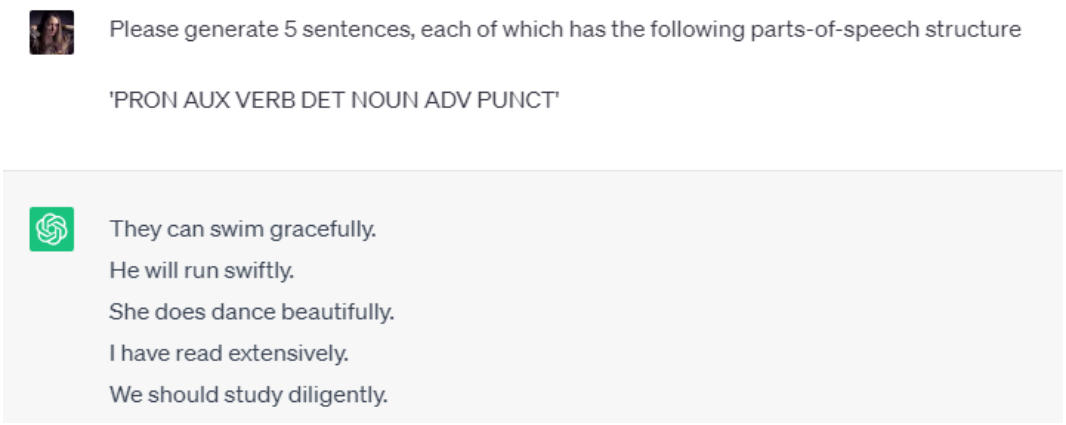
However, ChatGPT showed a large variety of generation, including the generation of different sentences corresponding to the same sequence of tags.

4.2.4 Remarks

It's also important to add some remarks about the evaluation process:

1. Evaluating the exact match for POS, sentiment and tense depends on the quality of the models that calculate these parameters after generation. Thus, such an evaluation may contain certain small deviations from actual values.

FIGURE 4.9: Example of prompt for generating sentences with ChatGPT based on given parts-of-speech structure



2. Perplexity metric is calculated under the GPT-2 model, hence it's by default biased in favor of GPT-2. Since in our work we use the perplexity metric to compare the performance of different fine-tuned GPT-2 models, we can rely on this metric to compare fluency. However, when comparing the results of the work with the results of the work of other models, it is worth to consider other fluency metrics, such as MAUVE [Pillutla et al., 2021] or human evaluation.

3. Generally speaking, comparing the results of GPT-2 and ChatGPT generations is possible only at the level of general conclusions from observations, since both models have not only different formulations of controllable generation tasks, but also different training backgrounds.

Chapter 5

Summary and future work

This study investigates possible directions of text generation with compliance with control conditions compliance. In our work with GPT-2, we were able to achieve decent results for controllable generation tasks with given length, parts-of-speech tags, sentiment, and tense control, as well as a combination of control conditions. Fine-tuned models not only learned to generate text under a certain condition, but also to produce fluent and diverse results even for complex or repetitive structures of parts-of-speech tags and combinations of different controls.

Diffusion-based approaches, despite the expected promise, haven't show good results on our dataset. However, our work with these models has laid the groundwork for further research in this direction, and we expect that with more resources and training time we will be able to achieve the expected results.

In our research, we identified the weaknesses of controllable text generation. In particular, the generation of texts of a certain length, especially for long sentences, is one of the difficulties faced by even the most complex language models. This gap can serve as another direction for future research.

An important component of controllable text generation, as well as any generation task, is training data. During our research, we realized that models that perform well on simple and narrow-topic datasets may not always perform well when trained on larger and more diverse datasets. However, researching the capabilities of models across a wide range of tasks and topics is important for understanding their generative capabilities beyond narrow domains.

In general, controllable text generation, like any actively emerging field, has many ways for development. The purpose of this work was to study some of the possible approaches, their possibilities and limitations. By investigating the facets of the controllable generation tasks, this research is hoping to contribute to the ongoing progress in this field.

Appendix A

Links to materials

A.1 Code

The public repository with code and generation results is available by the link:
https://github.com/SandraKonopatska/controllable-text-generation

A.2 Models

Fine-tuned GPT-2 models are available by the link:
https://drive.google.com/drive/folders/1Hr8rb_aeFn6HqPnAJq1C2O9gOQK86Ac1

A.3 Dataset

Dataset is available by the link:
https://drive.google.com/drive/folders/1KUKAHYEL9ZnibFaQrclhQBnjncT9xDRr

Appendix B

Evaluation metrics

FIGURE B.1: Evaluation metrics

Model	Number of sentences	Control metrics										
		Exact length	Appr length	Mean length deviation	Exact sentiment	Appr sentiment	Exact tense	Exact POS	Unique POS without order	2-grams POS	3-grams POS	Alignment POS
GPT-2 length	1000	42.3%	93.1%	1.551	-	-	-	-	-	-	-	-
GPT-2 length	2000	46.3%	93.2%	1.395	-	-	-	-	-	-	-	-
GPT-2 sentiment	1000	-	-	-	73.6%	99.1%	-	-	-	-	-	-
GPT-2 tense	1000	-	-	-	-	-	98.7%	-	-	-	-	-
GPT-2 POS different promts	1000	-	-	-	-	-	-	41.2%	97.8%	93.0%	87.4%	92.1%
GPT-2 POS promts with repetition	1000	-	-	-	-	-	-	39.3%	97.5%	92.3%	86.4%	92.4%
GPT-2 POS sentiment tense	1000	-	-	-	62.4%	97.7%	92.4%	40.9%	97.4%	92.1%	86.2%	92.6%
ChatGPT length	146	28.8%	87.0%	1.53	-	-	-	-	-	-	-	-
ChatGPT POS promts with repetition	50	-	-	-	-	-	-	0.0%	76.1%	41.7%	17.8%	49.2%

Model	Number of sentences	Fluency metrics	Distinctiveness metrics									
		Perplexity	1-grams in sentence	2-grams in sentence	3-grams in sentence	1-grams in all sentences	2-grams in all sentences	3-grams in all sentences	Distinct endings, 2 tokens	Distinct endings, 3 tokens	Distinct beginnings, 2 tokens	Distinct beginnings, 3 tokens
GPT-2 length	1000	164.694	91.0%	98.8%	99.6%	22.3%	73.7%	95.6%	76.1%	95.4%	76.9%	96.2%
GPT-2 length	2000	160.064	91.1%	98.8%	99.6%	16.8%	66.5%	93.0%	68.0%	96.0%	69.5%	92.3%
GPT-2 sentiment	1000	136.96	90.7%	98.6%	99.5%	22.3%	73.9%	95.7%	79.3%	98.0%	78.2%	95.7%
GPT-2 tense	1000	144.459	91.2%	98.8%	99.6%	22.0%	73.7%	95.5%	80.5%	96.6%	74.9%	95.1%
GPT-2 POS different promts	1000	324.028	92.1%	99.2%	99.8%	24.2%	78.2%	97.1%	77.1%	97.3%	80.4%	97.3%
GPT-2 POS promts with repetition	1000	313.777	91.3%	99.2%	99.8%	24.4%	77.0%	95.8%	76.7%	96.5%	80.4%	94.9%
GPT-2 POS sentiment tense	1000	354.1	92.0%	99.3%	99.8%	24.1%	78.5%	97.3%	77.8%	97.8%	79.1%	97.4%
ChatGPT length	146	56.416	91.3%	99.9%	100.0%	27.9%	62.0%	81.7%	93.8%	96.6%	35.6%	57.5%
ChatGPT POS promts with repetition	50	537.96	94.3%	99.7%	100.0%	57.8%	95.0%	100.0%	96.0%	100.0%	96.0%	100.0%

Model	Number of sentences	Distinctiveness metrics							Repetition metrics	
		Repeated POS 2-grams	Repeated POS 3-grams	Repeated POS 5-grams	Repeated POS endings, 2 tokens	Repeated POS endings, 3 tokens	Repeated POS beginnings, 2 tokens	Repeated POS beginnings, 3 tokens	3-grams	5-grams
GPT-2 length	1000	-	-	-	-	-	-	-	56.2%	96.2%
GPT-2 length	2000	-	-	-	-	-	-	-	56.3%	96.3%
GPT-2 sentiment	1000	-	-	-	-	-	-	-	55.2%	95.6%
GPT-2 tense	1000	-	-	-	-	-	-	-	55.4%	96.1%
GPT-2 POS different promts	1000	-	-	-	-	-	-	-	66.8%	98.1%
GPT-2 POS promts with repetition	1000	91.7%	97.4%	99.4%	91.0%	98.2%	90.0%	96.1%	67.4%	98.6%
GPT-2 POS sentiment tense	1000	-	-	-	-	-	-	-	67.8%	98.3%
ChatGPT length	146	-	-	-	-	-	-	-	-	-
ChatGPT POS promts with repetition	50	99.2%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	-	-

Bibliography

Austin, Jacob et al. (2021). *Structured Denoising Diffusion Models in Discrete State-Spaces*. DOI: 10.48550/ARXIV.2107.03006. URL: https://arxiv.org/abs/2107.03006.

Barbieri, Francesco et al. (2020). *TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classification*. arXiv: 2010.12421 [cs.CL].

Brown, Tom B. et al. (2020). *Language Models are Few-Shot Learners*. DOI: 10.48550/ARXIV.2005.14165. URL: https://arxiv.org/abs/2005.14165.

Chen, Nanxin et al. (2020). *WaveGrad: Estimating Gradients for Waveform Generation*. DOI: 10.48550/ARXIV.2009.00713. URL: https://arxiv.org/abs/2009.00713.

Chen, Ting, Ruixiang Zhang, and Geoffrey Hinton (2022). *Analog Bits: Generating Discrete Data using Diffusion Models with Self-Conditioning*. DOI: 10.48550/ARXIV.2208.04202. URL: https://arxiv.org/abs/2208.04202.

Chowdhery, Aakanksha et al. (2022). *PaLM: Scaling Language Modeling with Pathways*. DOI: 10.48550/ARXIV.2204.02311. URL: https://arxiv.org/abs/2204.02311.

Dathathri, Sumanth et al. (2019). *Plug and Play Language Models: A Simple Approach to Controlled Text Generation*. DOI: 10.48550/ARXIV.1912.02164. URL: https://arxiv.org/abs/1912.02164.

Deng, Yuntian et al. (2020). *Residual Energy-Based Models for Text Generation*. arXiv: 2004.11714 [cs.CL].

Ficler, Jessica and Yoav Goldberg (2017). *Controlling Linguistic Style Aspects in Neural Language Generation*. arXiv: 1707.02633 [cs.CL].

Gong, Shansan et al. (2022). *DiffuSeq: Sequence to Sequence Text Generation with Diffusion Models*. DOI: 10.48550/ARXIV.2210.08933. URL: https://arxiv.org/abs/2210.08933.

Gu, Jiatao et al. (2017). *Non-Autoregressive Neural Machine Translation*. DOI: 10.48550/ARXIV.1711.02281. URL: https://arxiv.org/abs/1711.02281.

Han, Xiaochuang, Sachin Kumar, and Yulia Tsvetkov (2022). *SSD-LM: Semi-autoregressive Simplex-based Diffusion Language Model for Text Generation and Modular Control*. DOI: 10.48550/ARXIV.2210.17432. URL: https://arxiv.org/abs/2210.17432.

Ho, Jonathan, Ajay Jain, and Pieter Abbeel (2020). *Denoising Diffusion Probabilistic Models*. DOI: 10.48550/ARXIV.2006.11239. URL: https://arxiv.org/abs/2006.11239.

Ho, Jonathan et al. (2022). *Video Diffusion Models*. DOI: 10.48550/ARXIV.2204.03458. URL: https://arxiv.org/abs/2204.03458.

Hoogeboom, Emiel et al. (2021a). *Argmax Flows and Multinomial Diffusion: Learning Categorical Distributions*. DOI: 10.48550/ARXIV.2102.05379. URL: https://arxiv.org/abs/2102.05379.

Hoogeboom, Emiel et al. (2021b). *Autoregressive Diffusion Models*. DOI: 10.48550/ARXIV.2110.02037. URL: https://arxiv.org/abs/2110.02037.

Hu, Zhiting et al. (2018). *Toward Controlled Generation of Text*. arXiv: 1703.00955 [cs.LG].

Jelinek, F. et al. (Aug. 2005). "Perplexity—a measure of the difficulty of speech recognition tasks". In: *The Journal of the Acoustical Society of America* 62.S1, S63–S63. ISSN: 0001-4966. DOI: 10.1121/1.2016299. eprint: https://pubs.aip.org/asa/jasa/article-pdf/62/S1/S63/11558910/s63_5_online.pdf. URL: https://doi.org/10.1121/1.2016299.

Keskar, Nitish Shirish et al. (2019). *CTRL: A Conditional Transformer Language Model for Controllable Generation*. DOI: 10.48550/ARXIV.1909.05858. URL: https://arxiv.org/abs/1909.05858.

Kingma, Diederik P. et al. (2021). *Variational Diffusion Models*. DOI: 10.48550/ARXIV.2107.00630. URL: https://arxiv.org/abs/2107.00630.

Kong, Zhifeng et al. (2020). *DiffWave: A Versatile Diffusion Model for Audio Synthesis*. DOI: 10.48550/ARXIV.2009.09761. URL: https://arxiv.org/abs/2009.09761.

Krause, Ben et al. (2020). *GeDi: Generative Discriminator Guided Sequence Generation*. DOI: 10.48550/ARXIV.2009.06367. URL: https://arxiv.org/abs/2009.06367.

Li, Jiwei et al. (2016). *A Persona-Based Neural Conversation Model*. arXiv: 1603.06155 [cs.CL].

Li, Xiang Lisa et al. (2022). *Diffusion-LM Improves Controllable Text Generation*. DOI: 10.48550/ARXIV.2205.14217. URL: https://arxiv.org/abs/2205.14217.

Liu, Alisa et al. (2021). *DExperts: Decoding-Time Controlled Text Generation with Experts and Anti-Experts*. DOI: 10.48550/ARXIV.2105.03023. URL: https://arxiv.org/abs/2105.03023.

Lovelace, Justin et al. (2022). *Latent Diffusion for Language Generation*. arXiv: 2212.09462 [cs.CL].

Needleman, Saul B. and Christian D. Wunsch (1970). "A general method applicable to the search for similarities in the amino acid sequence of two proteins". In: *Journal of Molecular Biology* 48.3, pp. 443–453. ISSN: 0022-2836. DOI: https://doi.org/10.1016/0022-2836(70)90057-4. URL: https://www.sciencedirect.com/science/article/pii/0022283670900574.

Nichol, Alex and Prafulla Dhariwal (2021). *Improved Denoising Diffusion Probabilistic Models*. DOI: 10.48550/ARXIV.2102.09672. URL: https://arxiv.org/abs/2102.09672.

Pillutla, Krishna et al. (2021). *MAUVE: Measuring the Gap Between Neural Text and Human Text using Divergence Frontiers*. arXiv: 2102.01454 [cs.CL].

Radford, Alec et al. (2019). "Language Models are Unsupervised Multitask Learners". In.

Ren, Yi et al. (2020). *A Study of Non-autoregressive Model for Sequence Generation*. DOI: 10.48550/ARXIV.2004.10454. URL: https://arxiv.org/abs/2004.10454.

Saharia, Chitwan et al. (2020). *Non-Autoregressive Machine Translation with Latent Alignments*. DOI: 10.48550/ARXIV.2004.07437. URL: https://arxiv.org/abs/2004.07437.

Scialom, Thomas et al. (2020). *Discriminative Adversarial Search for Abstractive Summarization*. arXiv: 2002.10375 [cs.CL].

Sohl-Dickstein, Jascha et al. (2015). *Deep Unsupervised Learning using Nonequilibrium Thermodynamics*. DOI: 10.48550/ARXIV.1503.03585. URL: https://arxiv.org/abs/1503.03585.

Sohn, Kihyuk, Honglak Lee, and Xinchen Yan (2015). "Learning Structured Output Representation using Deep Conditional Generative Models". In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes et al. Vol. 28. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2015/file/8d55a249e6baa5c06772297520da2051-Paper.pdf.

Song, Yang et al. (2020). *Score-Based Generative Modeling through Stochastic Differential Equations*. DOI: 10.48550/ARXIV.2011.13456. URL: https://arxiv.org/abs/2011.13456.

Strudel, Robin et al. (2022). *Self-conditioned Embedding Diffusion for Text Generation*. DOI: 10.48550/ARXIV.2211.04236. URL: https://arxiv.org/abs/2211.04236.

Tomczak, Jakub M (2022). *Deep Generative Modeling*. Springer Nature.

Wang, Ke and Xiaojun Wan (July 2018). "SentiGAN: Generating Sentimental Texts via Mixture Adversarial Networks". In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, pp. 4446–4452. DOI: 10.24963/ijcai.2018/618. URL: https://doi.org/10.24963/ijcai.2018/618.

Yang, Kevin and Dan Klein (2021). "FUDGE: Controlled Text Generation With Future Discriminators". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics. DOI: 10.18653/v1/2021.naacl-main.276.

Zhang, Hanqing et al. (2022a). *A Survey of Controllable Text Generation using Transformer-based Pre-trained Language Models*. DOI: 10.48550/ARXIV.2201.05337. URL: https://arxiv.org/abs/2201.05337.

Zhang, Susan et al. (2022b). *OPT: Open Pre-trained Transformer Language Models*. DOI: 10.48550/ARXIV.2205.01068. URL: https://arxiv.org/abs/2205.01068.

Zhao, Junbo, Michael Mathieu, and Yann LeCun (2017). *Energy-based Generative Adversarial Network*. arXiv: 1609.03126 [cs.LG].