A Plug-and-Play Approach to Age-Adaptive Dialogue Generation

Lennert Jansen
lennertjansen95@gmail.com
Thesis proposal

Contents

1 Research questions **2**

2 Introduction **3**

3 Literature review **3**
3.1 Background 3
3.2 Related work 6

4 Methodology **8**
4.1 Transformers 8
4.2 Causal language modeling with Transformers 9
4.3 Plug-and-play modeling 10

5 Planning **12**

1 Research questions

1. To what extent can a classifier automatically detect age-related linguistic differences in natural language? And which features are most helpful in age-group detection?
   (a) Do they (i.e., the linguistic or latent features exploited by the classifier) match the age-related informative features reported in previous work?
      - **Hypothesis:** I expect the age-group of a text's producer to be reliably detectable. Also, I suspect efficacy of the models to increase with their complexity, with complexity going from n-gram, to RNN-based, to Transformer-based. I suspect linguistic trends in age found in Pennebaker and Stone [2003] and Schler et al. [2006] to persist in the feature importance of the lexically focused n-gram models. Moreover, I expect the encoder-decoder architectures to uncover more latent age-related patterns in language that eluded earlier automated age-detection work, e.g., Nguyen et al. [2011]. Finally, I suspect the differences in language use between age-groups to lie more at the structural than lexical level.
      - **Implications:** This question aims to lay the foundation of the later research objectives by first establishing whether a discriminator can successfully categorize discourse into correct user age-brackets. I plan to build increasingly complex classifiers, starting from a bag-of-words (BoW) n-gram model, progressing to embedding-based architectures (LSTM and BERT). The next entailed goal is to investigate which aspects of users' language use seem to play a role in determining the corresponding age groups. NB: these salient characteristics could also express themselves in the latent spaces of neural classifiers. Finding and understanding such features will not only help me better understand the relationship between age-group and word use, but also provide guidelines for the development of controllable conversational response generators later in the project.
2. Can large-scale language models (LMs), i.e. GPT-2 [Radford et al., 2019] or DialoGPT [Zhang et al., 2020], be leveraged for text generation, controlled for age-groups?
   (a) What role does the used data play in the differences in output and performance between the base (conversational) language models and their controlled counterparts?
      - **Hypothesis:** Based on Madotto et al. [2020], Dathathri et al. [2020], Pennebaker and Stone [2003], I expect they can. Assuming that age-group is distinctive enough a trait of language use for it to notably affect writing style when controlled for (see previous hypothesis), age can be seen as an attribute input that neatly fits in the the plug-and-play setup proposed by Dathathri et al.. My research is conducted using **(1)** a dataset of blogs labeled by author age, and **(2)** a collection of transcribed conversations labeled by speaker age. I do not expect the controllability of the guided generation model to be impacted by the fact that one is conversational and the other is not. The ways that I do expect the choice of data for the controllable system to impact its performance and implications are differences in the mode of discourse (written versus spoken), recency (blogs are from 2004, conversations from 2014), and input length. Finally, it is to be expected from Dathathri et al. [2020], Madotto et al. [2020] that the fluency of the base language model will tend to deteriorate after controlling for either dataset.
      - **Implications:** To confirm the PPLM setup with age as an attribute input. First step in achieving adaptive dialogue generation, as non-conversational text generation is less restrictive.
3. To what degree is such a controlled text generation (CTG) model successful in dialogue that is adaptive w.r.t. age, such that it has a detectable effect on the perception of the user?
   - **Hypothesis:** I expect the perceived quality of the dialogues with the adaptive system to be adequate, but decrease for increasing numbers of conversational turns with a user. It can also be expected that evaluating if generated responses are representative of age-specific vernacular will be more decisive with automatic than human evaluation. Nevertheless, I expect the age-differences in generated language to be noticeable by humans.
   - **Implications:** This is in alignment with the objectives of the Human(e) AI research project. The ultimate challenge of this thesis will be to leverage the grammatical fluency of GPT-2 or DialoGPT and the inexpensive tunability of Dathathri et al.'s PPLM and Madotto et al. [2020]'s PPCM setups for age-adaptive neural conversational response generation.

2 Introduction

In recent years, we have witnessed promising advances in downstream natural language processing (NLP) tasks, such as language modeling, reading comprehension, machine translation, controllable text generation, and conversational response generation [Radford et al., 2019, Bahdanau et al., 2015, Dathathri et al., 2020, Madotto et al., 2020]. Vaswani et al. [2017]'s Transformer architecture plays a central role in many of the state of the art (SotA) solutions to these problems, emerging as the succession to recurrent neural network (RNN) models for NLP tasks. Transformer-based language models (LMs) pre-trained on massive amounts of textual data, most famously OpenAI's GPT-2 [Radford et al., 2019], have demonstrated their usefulness for several of the aforementioned NLP tasks. For instance, controllable text generation and producing dialogue responses have improved greatly because of GPT-based hybrid models. Given the non-trivial requirements for sufficiently massive training corpora and computational resources, GPT-based solutions are worth considering for these two problems.

Controllable text generation (CTG) tries to enforce abstract properties, like writing style, on the passages being produced. Fine-tuning large-scale LMs for writing-style adaptation is extremely expensive, but Dathathri et al. [2020] and Li et al. [2020] propose methods that both excel at the task, while bypassing significant retraining costs. Dialogue response generation is the task of producing replies to a conversational agent's prompts, in a manner that is ideally both non-repetitive and relevant to the course of the conversation. With DialoGPT, Zhang et al. [2020] also manage to leverage GPT-2's powerful fluency for dialogue tasks, by framing them as language modeling tasks where multi-turn dialogue sessions are seen as long texts.

CTG and dialogue response generation share the overarching objective of producing grammatically correct text that is distinct from any training instance. A blend of these two tasks, i.e., controllable dialogue response generation, is an interesting and only partially explored route. It ties closely to one of Artificial Intelligence's long-standing goals of achieving human-like conversation with machines, as humans are known to adapt their language use to the characteristics of their interlocutor [Gallois and Giles, 2015]. Adaptive dialogue generation is difficult due to the challenge of representing traits, like age or gender, via language expression, and the lack of persona trait labeled dialogue datasets. Zheng et al. [2019] explore the problem of personalized dialogue generation, and introduced `PersonalDialog` a large-scale multi-turn Chinese Mandarin dialogue dataset with personality trait labeling, and persona-aware adaptive dialogue generation models using RNNs and attention mechanisms.

In this thesis, I investigate the problem of controllable dialogue generation, with a focus on adapting responses to users' age. As a preliminary research objective, I aim to study to what extent a classifier can detect age-related linguistic differences in natural language And which features are most helpful in age-group detection? Do they (i.e., the linguistic or latent features exploited by the classifier) match the age-related informative features reported in previous work? After empirically confirming that speaker age detection is possible, I explore whether large-scale LMs, e.g. GPT-2, can be leveraged for text generation, controlled for age-groups. And what role does the used data play in the differences in output and performance between regular GPT-2 and controllable GPT-2? Finally, and most importantly, my research focuses on the degree to which such a CTG model is successful in generating dialogue that is adaptive w.r.t. age, such that it has a detectable effect on the perception of the user.

The remainder of this report is structured as follows: Sub-section 3.1 positions the subject of controlled text generation in its theoretical background, and and 3.2 compares it to the most relevant related work. The methodology in Section 4 gives detailed explanations of the most important modeling methods and techniques used for this research. And Section 5 provides a timetable for the remainder of this project.

3 Literature review

3.1 Background

Controllable dialogue generation encompasses several concepts in natural language processing and linguistics that must be understood to approach the problem. This subsection highlights these topics and positions the central problem of this thesis in its relevant theoretical background.

Language Models Language modelling is central to many NLP tasks. A language model (LM) is a probability distribution over words in a sentence or document. They are trained to predict the probability of the next word in an sentence, given the preceding sequence of words. The language modelling task is formulated as an unsupervised distribution estimation problem of datapoints $\{\mathbf{x}_1, ..., \mathbf{x}_N\}$ (e.g., documents), each representing sequences (of e.g., symbols or tokens) of varying lengths $(s_{i,1}, ..., s_{i,n}), i \in \{1, ..., N\}$. Note that N denotes the corpus size, and n the sequence length of datapoint i. To avoid cluttered notation, the subscript i will sometimes be omitted when discussing an arbitrary datapoint. The probability distribution over an observation $\mathbf{x}$ (i.e., the joint probability of an ordered sequence) can then be factorised as the product of its constituent conditionals [Radford et al., 2019]:

$$p(\mathbf{x}) = \prod_{j=1}^{n} p(s_j|s_1, ..., s_{j-1}). \tag{1}$$

This task allows language models to detect and learn patterns in language. The learned representations of these patterns can then be used for a plethora of applications, such as classification, and text generation. Moreover, this results in a framework for tractable sampling from the unconditional language model $p(\mathbf{x})$. $p(\mathbf{x})$ can therefore be seen a base generative model that can generate sample sentences [Dathathri et al., 2020].

In recent years, the attention-based models, Transformers [Vaswani et al., 2017], have replaced recurrent neural networks (RNNs) as the dominant architecture for LMs, with major improvements in distribution estimation, long-range dependency handling, sample diversity, and parallel processing. Another recent development in language modelling is that of pre-training LMs on massive corpora. So-called large-scale general purpose LMs have demonstrated significant improvements in downstream tasks, i.e., other NLP tasks for which the model was not specifically trained or fine-tuned. Most famously the OpenAI's series of Generative Pre-trained Transformer (GPT) models have improved numerous NLP benchmarks [Radford et al., 2018, 2019, Brown et al., 2020].

(Controllable) Text Generation In text generation, a LM $p(\mathbf{x})$ is asked to produce text $\mathbf{x}$ given a primer text by using the language model to sample from the distribution of words that are assigned the highest likelihood of following the primer text. Text generation in itself is the task of generating a piece of text given an input text. This process can be seen as sampling from a conditional distribution. Controllable text generation refers to the more restrictive problem of enforcing higher-level linguistic features on the generated text during sampling. This can be seen as a sub-problem of vanilla text generation, because the conditioning factor for the output text is further constrained to also include some predefined textual attribute. This attribute can be many things, like sentiment, topic, or writing style.

Controllable text generation or CTG is a more challenging problem than vanilla text generation for a number of reasons. First, defining the desired attribute to be controlled for in a manner that is intelligible for a machine is a challenge in self. Second, like many and NLP problems, there do not exist many parallel corpora for which accurate attribute labeling is available. Furthermore, the measure of attribute adherence is a very vague and ambiguous concept. Namely, a text can be written in an extremely positive sentiment in multiple formulations, all of which adhere to the positive sentiment. Another important hurdle for controllable text generation, especially when CTG is combined to leverage the linguistic power of large-scale language models, is that the cost of fine-tuning or pre-training a model to control for a linguistic attribute can be very high.

PPLM [Dathathri et al., 2020] is an example of a recent work that has the primary focus of leveraging powerful large-scale language models and making them controllable for a wide variety of linguistic attributes, all while avoiding incurring significant costs of fine-tuning. Nevertheless avoiding these costs is anything but trivial. The plug and play set up of the PPLM model forms one of the main theoretical foundations of this work. It is both logical and promising for every day engineers to be able to leverage the grammatical fluency of pre-trained language models for more specific downstream tasks, e.g., specifying linguistic characteristic to enforce on an automatically written text. Their setup consist of a symbiosis of GPT-2 as their powerful large-scale language model, and a significantly smaller and therefore easier to train and fine-tune attribute model. This attribute model is often a small classifier, and can range in complexity from a simple bag of words model with a logistic classifier to a more complicated transformer encoder head. The main benefit of this setup is the extensibility it brings with it. Namely, such large-scale language models are open-source and available online and can be tailored to their specific needs using a significantly easier to train attribute model of your own liking. Dathathri et al. [2020] demonstrate the applicability of their model on a wide variety of tasks ranging from text style transfer to language detoxification (all of which can be seen as sub problems of controllable text generation). Other real-world applications include: being able to automatically re-write or adjust a draft text for an editorial, automatic generation of brand specific vacancy ads, or personalized chatbot assistance or even personalized education. What this work also provides is a starting point for new applications, namely controllable dialogue generation.

Dialogue response generation The domain of text generation encapsulates a number of sub-tasks, e.g., machine translation, abstractive summarization, and paraphrasing. Dialogue response generation is also a special case of language generation. It can be seen as language generation where the prompt is a turn in a dialogue session. Conversational response generation shares open-domain text generation's overarching objective of producing grammatically correct fluent text that is distinct from any training instance, while remaining relevant to the prompt. However, computational dialogue modeling distinguishes itself from most NLP domains due to the challenges associated with modeling human conversation: informal, noisy, unstructured, and even erroneous real-world responses, possibly competing goals of interlocutors, or an inherently more diverse set of acceptable responses. The last point also emphasizes that neural response generation has a much more 1-*to-many* nature than most text generation tasks.

Despite these differences, conversational response generation can be modeled in similar ways to open-domain text generation. Zeng et al. [2020] suggest to either formulate it in terms of source-target pairs, much like neural machine translation, or as a language modeling objective, where the next token or utterance is conditioned on the dialogue history. To remain close to the training objectives of my baseline models (GPT-2 [Radford et al., 2019] and DialoGPT [Zhang et al., 2020]) I choose to adopt the language modeling formulation for conversation modeling. I.e., concatenate all dialogue turns in a multi-turn dialogue session into a long text: $x_1, ..., x_N$. Denote the source sentence or dialogue history as $S = x_1, ..., x_m$ and the target sentence (ground truth response) as $T = x_{m+1}, ..., x_N$. The conditional probability of dialogue continuation given its history $P(T|S)$ can be written as

$$p(T|S) = \prod_{n=m+1}^{N} p(x_n|x_1, ..., x_{n-1}). \tag{2}$$

A multi-turn dialogue session $T_1, ..., T_K$ can be written as $p(T_K, ..., T_2|T_1)$ which is essentially the product of all source-target pairs probabilities $p(T_i|T_1, ..., T_{i-1})$. This formulation also shows that optimising the single objective $p(T_K, ..., T_2|T_1)$ is equivalent to optimising all source-target pair probabilities.

Controllable dialogue generation Endowing a dialogue system with personality traits to generate human-like conversation is a long-standing goal in AI. This objective is difficult to reach because of the challenge of representing personality traits via language expression and the lack of large-scale persona-labeled dialogue datasets [Zheng et al., 2019]. Assuming an encoder-decoder setup, Zheng et al. [2019] argue that most personalized neural conversation models can be classified as one of two types: implicit and explicit personalisation models. For implicit personalization models, each speaker has its own vector representation, which implicitly captures the speaking style of the speaker in the decoding process. These models enjoy the benefit of having a more granular and realistic representation of speaking style, as opposed to a simple discrete set of traits (as is the case for explicit personalization models). On the other hand, it is unclear how speaker style is captured and should be interpreted, as all the information about a speaker's style is encoded in a real-valued vector. Furthermore, these methods suffer from a data sparsity issue, because each dialogue should be tagged with a speaker identifier and there should be sufficient dialogues from each trait-group to train a reliable trait-adaptive model. This last drawback is a bigger hurdle for the Zheng et al. [2019] than it is for mine, as their work deals with personalization for intersections of multiple traits, whereas my work focuses on adaptation to a small number of age groups.

For explicit personalization models, the generated responses are conditioned either on a given personal profile, text-described persona, or simply an attribute label. I.e., speaker traits are represented as key-value pairs or descriptions about age, gender, etc. This can be seen as conditioning the decoder's output on an attribute a, much like the PPLM setup of Dathathri et al. [2020]. Speakers with same set of personality traits can share attribute representations, so it does not require a speaker-specific representation vector. Such structured character descriptions are more explicit, straight-forward, and interpretable. However, explicit personalization models require manually labeled or crowdsourced datasets for development, making it difficult to scale these models to large-scale dialogue datasets.

Language and age. The relationship between a person's age and use of language is a thoroughly studied subject with a decades long history and inherent challenges [Pennebaker and Stone, 2003, Nguyen et al., 2014]. A number factors like community membership (e.g., gender, socioeconomic status, or political affiliation), experimental condition (e.g., emotional versus non-emotional disclosure), mode of disclosure (writing versus talking), and other confounding variables complicate the study of age's relation to language [Nguyen et al., 2011]. The relatively recent advent of widely available computational resources and vast amounts of textual data made it possible to leverage machine learning methods to help detect patterns in language that eluded conventional sociolinguistic research. Early computational investigations into the connection between a person's age and use of language is typically a combination of qualitative and statistical methods. For instance, using a mix between their proprietary count-based text analysis framework, Linguistic Inquiry and Word Count (LIWC) and sociolinguistic theory, Pennebaker and Stone [2003] study the changes in written and spoken language use with increasing age. They discuss four important areas of a person's character that have been found to change with age: emotional experience and expression, identity and social relationships, time orientation, and cognitive abilities. These four axes and their hypothesized relationships with language use and age can be interpreted in the following ways:

1. *Emotional experience and expression:* This is the relationship between increasing age and linguistically observable manifestations of a person's experienced emotions. In practical terms, this is framed as detectable instances of positive and negative affect in language. This complex relationship between age and emotional expression is characterized by decreased levels of negative affect and slightly non-decreasing levels of positive affect. This is also confirmed by the findings of Schler et al. [2006].

2. *Sense of identity and social relationships:* These terms refer to developmental tends in one's relation to self and others, as expressed in their language, e.g., as references to self (*I, me, my*, and *we, us, our*) or others (*they, them, theirs*). The literature cited in Pennebaker and Stone [2003] suggests that the *quantity* of social connections decreases and the *quality* of remaining relationships increases with age.

3. *Time orientation:* This relationship describes how people express their perception of and orientation towards time. For instance, this can be indicated by the use of time-related verb tenses. The authors suggest that older individuals tend to be more past-oriented than their younger future-oriented counterparts.

4. *Cognitive abilities:* This refers to markers of cognitive capacity in language. Aging is expected to be associated with less use of cognitively complex words after a certain mid-adulthood peak. Specifically, the relationship between markers of cognitive complexity in natural language (cognitive mechanisms, causal insight, and exclusive words) and age is hypothesized to be curvilinear. And because verbal ability does not decline until very late in life, markers of verbal ability (e.g., use of big words) are not expected to show changes with age.

Pennebaker and Stone [2003] consider the following variables: positive and negative emotions, first-person singular and first-person plural pronouns, social references, time-related words (past-tense, present-tense, and future-tense verbs), big words (> 6 letters), cognitive mechanisms, causal insight, and exclusive words. Their main findings suggest that increasing with age, people use more positive and fewer negative affect words, use fewer self-references, use more future-tense and fewer past-tense verbs, and exhibit a general pattern of increasing cognitive complexity.

Detectable linguistic differences between age-groups can often be attributed to the use of language fads or references to age-specific popular culture. For instance, Schler et al. [2006] find that the use of slang and neologisms (such as *lol* and *ur*) are strong indicators of youth. Similarly, words like 'facebook', 'instagram', and 'netflix' appear in the most frequently used words by younger participants of conversational data collection efforts, like that of the British National Corpus' spoken component [Love et al., 2017].

More recent studies, like that of Nguyen et al. [2011] and Abdallah et al. [2020], frame age prediction from text as traditional machine learning problems, like linear regression, support vector machines, or neural architectures. These modeling approaches tend to reveal that strong indicators of age lie at the syntactic or structural level of language use, as opposed to the more content-based lexical level. Furthermore, this could explain why automatic detection from text of more content-based traits, like topic or sentiment, tend to be easier problems to solve than age prediction from text. To emphasize one such complicating factor, Nguyen et al. [2014] argue that differences in language use are often relation to the speaker's social identity, which could differ from their biological identity. This idea that age prediction from text is more challenging than topic or sentiment prediction could be an indication that controlled language generation for age-differences is also a more nuanced problem than topical steered text generation.

3.2 Related work

Controllable language generation Dathathri et al. [2020] achieve controllable language generation using a plug-and-play model set up. Their architecture uses GPT-2 to as their base language model which provides grammatical fluency, combined with a significantly easier to train attribute model (i.e., a simple BoW or single-layer classifier). Using gradient updates to the activation space from the much smaller attribute model they manage to generate language that combines (some of) the fluency of GPT-2 with the stylistic control of the attribute model, without the cost of retraining a specialised architecture. They demonstrate desirable fluency as measured by perplexity with respect to GPT [Radford et al., 2018] as well as measurable attribute control. Their architecture's applicability is also demonstrated on tasks such as controlled story writing and language detoxification. They also show a clear trade-off between attribute control and grammatical correctness and diversity. Previous, non-plug-and-play approaches to controlled language generation require fine-tuning large language models or training conditional generative LMs from scratch. Most notably CTRL [Keskar et al., 2019], which achieves controllable generation by training a generative Transformer for a number of control codes. CTG models that are not plug-and-play, like CTRL, can produce high quality fluent because they are specifically trained to maximize $p(\mathbf{x}|a)$, but require training massive language models with computational costs infeasible for most developers.

Recent examples of controllable language generation models that are not Transformer-based also exist. Li et al. [2020] introduce OPTIMUS, a large pre-trained Variational Autoencoder (VAE) Kingma and Welling [2014] that can be fine-tuned for specific natural language tasks, like guided sentence generation. They demonstrate OPTIMUS' ability to perform controlled text generation for from latent style embeddings, with fluency at par with GPT-2. They also show how OPTIMUS generalizes better for low-resource languages than BERT [Devlin et al., 2018]. Nevertheless, much like the previously mentioned non-plug-and-play CTG models, OPTIMUS still incurs a significant computational cost for fine-tuning per NLP task.

Writing-style Transfer Writing-style transfer is a closely related problem to controllable language generation. Its similarity lies in trying to modify the output distribution of a language generating model, such to control for abstract stylistic components of the produced text, keeping content and fluency preserved. It involves rewriting an input text with a different style of writing than it originally had. More formally, given a text $\mathbf{x}$, its corresponding style-representing vector $\mathbf{s}^{(i)}$, the number of different styles K over which there exists a distribution, and a desired style $\hat{\mathbf{s}} \in \{\mathbf{s}^{(i)}\}_{i=1}^{K}$, the goal of style transfer is to rewrite the input sentence to $\hat{\mathbf{x}}$ with style $\hat{\mathbf{s}}$, while preserving the information content of $\mathbf{x}$. Writing-style transfer can also be seen as a special case of (abstractive) summarization, for which Transformers have also demonstrated applicability [Baan et al., 2019].

Dai et al. [2019] introduce the Style Transformer, an alternative to the previous RNN-based encoder-decoder frameworks for text style transfer. In previous work, neural text style transfer was done by passing input text through an encoder, yielding a style-dependent latent representation $\mathbf{z}$. These previous approaches then attempted to "disentangle" $\mathbf{z}$ into a style-independent content representation and a latent representation of the stylistic properties of the input text. The subsequent decoder then receives the content representation and a new latent style variable as input, to ultimately produce a style-altered output text with unchanged content. This style-disentanglement approach has a number of drawbacks: **(1)** It is difficult to evaluate the quality of disentanglement of the latent space. **(2)** Disentanglement is unnecessary, as contemporary work by Lample et al. [2019] has shown a good decoder can perform controllable text generation from an entangled latent representation by "overwriting" the original style. **(3)** It is hard to capture rich semantic information in the latent representation due to limited capacity of vector representations (especially for long texts). **(4)** To disentangle style and content in the latent representations, all previous approaches have to assume all input texts can be encoded by a fixed-size latent vector. **(5)** Since most previous approaches use RNN-based encoder-decoder frameworks, they have problems capturing long-range dependencies in the input sentences.

These problems are addressed by Dai et al. [2019] using Transformers [Vaswani et al., 2017] as the building block for text style transfer. The authors' approach does not require any manipulation (i.e., disentanglement) of the latent space, eliminates the need for a fixed-size vector representation of the input, and handles long-range dependencies better due to Transformers' attention mechanism. Aside from this being the first application of Transformers for text style transfer, the work contributes a novel training algorithm for such models, and boasts significant improvements of results on two text style transfer datasets.

Dialogue Generation Dialogue generation is the text generation task of automatically generating a response given user's prompt. It is the essential precursor to this thesis' ultimate task of controllable dialogue generation. Zhang et al. [2020] introduce DialoGPT, a tunable large-scale language model for generation of conversational responses, trained on Reddit discussion chain data. DialoGPT therefore extends GPT-2 [Radford et al., 2019] to address a more restrictive sub-category of text generation, i.e., conversational response generation. DialoGPT inherits from GPT-2 a 12-to-48 layer transformer with layer normalization, a custom initialization scheme that accounts for model depth, and byte pair encodings [Sennrich et al., 2016] as a tokenizer. The generation task remains framed as language modeling, where a multi-turn dialogue session is modeled as a long text.

To address the well-known problem of open-domain text generation models producing bland and uninformative samples, Zhang et al. [2020] implement a maximum mutual information (MMI) scoring function. MMI uses a pre-trained backward model to predict $p(\texttt{source}|\texttt{target})$: i.e., the source sentences (dialogue history) given the target (responses, dialogue continuation). First, top-K sampling is used to generate a set of hypotheses. Then the probability $p(\texttt{source}|\texttt{hypothesis})$ is used to re-rank all hypotheses. As frequent and repetitive hypotheses can be associated with many possible queries/sources (i.e., a hypothesis that frequently occurs is one that is apparently applicable to many queries), and maximizing backward model likelihood penalizes repetitive hypotheses, MMI yields a lower probability for any specific query, thereby reducing blandness and promoting diversity.

DialoGPT is evaluated on the Dialog System Technology Challenge (DSTC) 7 track, an end-to-end conversational modeling task in which the goal is to generate conversation responses that go beyond chitchat by injecting information that is grounded in external knowledge. The model achieves state-of-the-art results on both the human and automatic evaluation results, by achieving near human-like responses that are diverse, relevant to the prompt, much like GPT-2 for open-domain language generation. They train 3 models of parameter sizes 117M, 345M, and 762M. The medium-sized 345M model achieves the best automatic evaluation results across most metrics, and is used as one of the baselines in later experiments in this thesis. Their Hugging Face PyTorch implementation can be tested here: `https://huggingface.co/microsoft/DialoGPT-medium`.

Controlled Dialogue Generation Controlled dialogue generation is the task of steering automatically generated conversational responses to contain certain desired stylistic aspects, like sentiment, specific topic, or more abstract writing-style characteristics. Nowadays, dialogue systems like Siri, Alexa, or Google Assistant, play large roles making technology easier to use, it is of great commercial interest to be able to control (e.g., personalize) the style of their responses. Medical applications too have been found for controllable dialogue generation. Zeng et al. [2020] explore

the applications of fine-tuning large language models, like GPT, on (Mandarin and English) medical consultation data. The resulting dialogue systems succeed at generating clinically correct and human-like responses to patients' medical questions. Medical dialogue systems like these can help to make healthcare services more accessible and aid medical doctors to improve patient care.

Zheng et al. [2019] investigate the problem of incorporating explicit personal characteristics in dialogue generation to deliver personalized conversation. Their main contributions are: **(1)** `PersonalDialog`, which is a large-scale multi-turn dialogue dataset with personality trait labeling (i.e., `Age`, `Gender`, `Location`, `Interest Tags`, etc.) for a large number of speakers. And **(2)** they propose persona-aware models that apply a trait fusion module in the encoder-decoder framework to capture and address personality traits in dialogue generation. Persona-aware attention mechanisms and bias are used to incorporate personality information in the decoding process. All their tested classification and dialogue generation models are either variations of RNNs (LSTMs or gated recurrent units (GRUs)), convolutional neural networks (CNNs), or hybrids of these model-classes (LSTM-outputs fed into a CNN, known as recurrent convolutional neural networks (RCNNs)). The authors study the influence of age, gender, and location on dialogue classification and generation, and use both automatic (perplexity, trait accuracy, and generated response diversity measures) and human evaluation. They find dialogues to be distinguishable by gender (about 90.61% test accuracy), then age (78.32% test accuracy), and finally location (62.04% test accuracy). Both automatic and human evaluation of the generated responses show that the best performing models benefit greatly from the persona-aware attention mechanism, possibly making a case to consider more attention-based architectures instead of RNNs.

Although the previously mentioned architectures manage to produce human-like fluent conversational responses, sometimes even leveraging the fluency of large pretrained LMs, they all suffer from the same computational drawback. They all require massive amounts of computational power to adapt their language styles, because in these cases guided generation implies fine-tuning (or even retraining) large attribute-specific dialogue datasets. For general controlled language generation, this obstacle is overcome by Dathathri et al. [2020]'s previously mentioned PPLM setup. The conversational analog of this idea, plug-and-play conversational model (PPCM), is proposed by Madotto et al. [2020]. Similar to PPLM, PPCM achieves guided dialogue generation via activation-space perturbations using easy to train attribute models. In this configuration, they can achieve controllable response generation without needing dialogue data or having to fine-tune a large language model. However, due to the computational complexity of PPLM's decoding process, it is unusable as practical conversational system. PPCM solves this problem by using residual adapters [Bapna and Firat, 2019] to tweak the decoding procedure such that it does not require more computation time. See Section 4.3 for a detailed explanation of the mechanisms behind PPLM and PPCM. Madotto et al. [2020] show, using both human and automatic evaluation, that PPCM can balance grammatical fluency and high degrees of attribute-adherence in its generated responses. PPCM uses DialoGPT as its base language model, and is tested for topical or sentimental attributes (i.e., positive, negative, sports, business, or science & tech). As mentioned in Section 3.1, I suspect these topical attributes to be more distinguishable at a lexical level than age-specific linguistic differences. My work therefore tries to extend the applicability of plug-and-play controlled generation to more abstract linguistic characteristics, using more complex attribute model architectures than the linear classifiers used by Dathathri et al. [2020] and Madotto et al. [2020].

4 Methodology

4.1 Transformers

The Transformer architecture plays a central role in most of the recent advances in NLP. The same holds for the methods used in this thesis to investigate controlled dialogue generation and speaker/author age detection. A brief explanation about the Transformer therefore in order. For a more detailed review of the model architecture, the reader is referred to the original paper ([Vaswani et al., 2017]) or this excellent blog post: `https://jalammar.github.io/illustrated-transformer/`.

The Transformer, like most neural sequence processing models, has an encoder-decoder structure. On a high level, the encoder receives an input sequence $\mathbf{x} = (x_1, ..., x_n)$ (e.g., a sentence), and maps this to a sequence of latent continuous variables $\mathbf{z} = (z_1, ..., z_n)$. The decoder then takes $\mathbf{z}$ as input, and maps this to an output sequence $\mathbf{y} = (y_1, ..., y_m)$. Note that the use of positional encodings of the input and output embeddings enables the Transformer to process and generate sequences in arbitrary order, allowing for a high degree of parallelization. The generation of $\mathbf{y}$ happens element-by-element in an auto-regressive fashion, where at step t, element y_{t-1} is also taken as input.

Both the encoder and decoder are comprised of N identical layers (denoted by the 'N $\times$' in the left part of Figure 1). Every sub-layer performs a succession of transformations using multi-head self-attention mechanisms and point-wise, fully connected layers, along with residual connections [He et al., 2016] around every sub-layer followed by layer normalization [Ba et al., 2016]. The decoder's first self-attention sub-layer is masked to ensure that the output predictions at sequence position i cannot depend on output positions greater than i. Finally, the decoder passes its output through a

linear and softmax layer to produce a probability distribution over the problem space (e.g., the vocabulary) from which the most likely symbols for the generated output sequence **y** can be sampled.

A key aspect of the Transformer architecture is its use of attention [Bahdanau et al., 2015]. This allows the encoder-decoder architecture to selectively focus on parts of the input sequence to produce a more informative hidden representation. Vaswani et al. formulate an attention function as a mapping of queries and sets of key-value pairs to an attention output, where matrices represent the queries Q, keys K, and values V. The attention output is a weighted sum of the values, based on the relevance of the corresponding keys to a query. In particular, they employ scaled dot-product attention:

$$\texttt{Attention}(Q, K, V) = \texttt{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \tag{3}$$

Furthermore, Vaswani et al. [2017] propose to use multi-head attention by using learned linear projections to project the queries, keys and values h times, and apply the aforementioned attention function to these projections in parallel. The concatenation of these attention outputs, passed through a linear layer, ultimately produces the final output of the Transformer's attention sub-layers. This allows the model to attend to the relevant information from all representation sub-spaces at various sequence positions. See Figure 1 for an schematic illustration of the Transformer's structure described above.

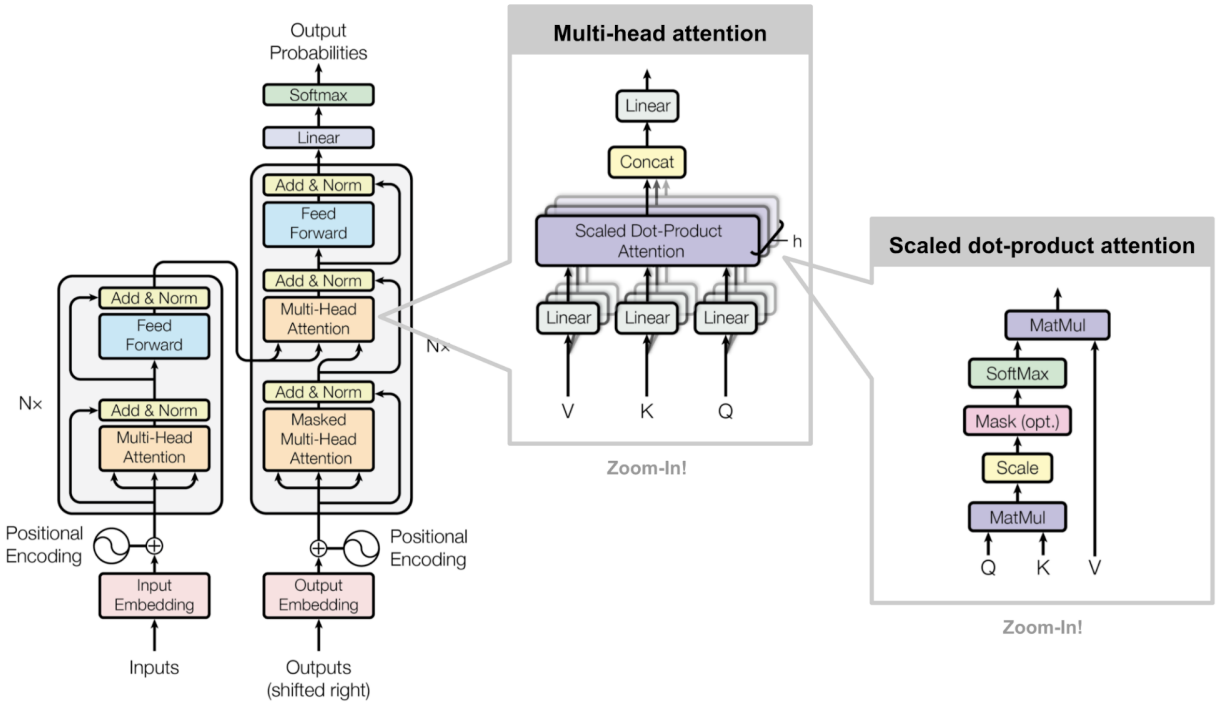

Figure 1: An overview of the full Transformer model architecture. *Collated image source:* Fig. 17 in this blog post `https://lilianweng.github.io/lil-log/2018/06/24/attention-attention.html`. *Original image source:* Figures 1 and 2 in Vaswani et al. [2017]

4.2 Causal language modeling with Transformers

Following the conventions of Dathathri et al. [2020] and Madotto et al. [2020], a dialogue is comprised of multiple alternating turns (sometimes referred to as utterances) between more than one speaker. For simplicity, this project only focuses on dialogues between two speakers. The conversation history at turn t is defined as $\mathcal{D}_t = \{S_1^{(1)}, S_1^{(2)}, ..., S_t^{(1)}\}$, where $S_t^{(j)}$ is speaker j's utterance at time t. Madotto et al. [2020] denote speaker 1 as the user U, and speaker 2 as the conversational system S, yielding dialogue history $\mathcal{D}_t = \{U_1, S_1, ..., U_t\}$. This notational convention will also be used for the user-system experiments later on in this report.

A Transformer-based language model (denoted `LM`) is used in this thesis to model the distribution of dialogues, using dialogue history at time t, $\mathcal{D}_t$, as a prompt to auto-regressively generate the dialogue continuation S_t. More specifically,

let the concatenation of the dialogue history at t and its continuation, $\{\mathcal{D}_t, S_t\}$, be represented as a sequence of tokens $\mathbf{x} = \{x_0, ..., x_n\}$. Then, by recursively applying the product rule of probability (Bishop [2006]), the unconditional probability of the sequence $p(\mathbf{x})$ can be expressed as:

$$p(\mathbf{x}) = \prod_{i=1}^{n} p(x_i|x_0, ..., x_{i-1}). \tag{4}$$

Dathathri et al. [2020] and Madotto et al. [2020] define the Transformer's decoding process in a recursive fashion. Let H_t denote the conversation history's key-value pairs, i.e., $H_t = \left[(K_t^{(1)}, V_t^{(1)}), ..., (K_t^{(l)}, V_t^{(l)})\right]$, with $(K_t^{(i)}, V_t^{(i)})$ representing the key-value pairs from the LM's i-th layer generated at all time steps 0 through t. This results in the recurrent dedocing process being expressed as:

$$o_{t+1}, H_{t+1} = \texttt{LM}\left(x_t, H_t\right), \tag{5}$$

where o_{t+1} is the hidden state of the last layer. Finally, after applying a softmax transformation, the next token x_{t+1} is sampled from the resulting probability distribution, i.e., $x_{t+1} \sim p_{t+1} = \texttt{softmax}\left(W o_{t+1}\right)$, where W is a linear mapping from the model's last hidden state to a vector of vocabulary size. This recursive formulation allows for efficient text generation by leveraging cached memories, without repeated forward passes.

4.3 Plug-and-play modeling

Plug-and-play language model (PPLM) Dathathri et al. [2020] works by using a text classifier, referred to as an attribute model, to control the text generated by a language model. Let $p(X)$ denote the distribution of a Transformer-based language model (e.g., GPT-2 or DialoGPT), where X represents the generated text. And $p(a|X)$ denotes the attribute model (e.g., a single-layer or BoW classifier) that represents the degree of adherence of text X to a certain attribute a (e.g., style, sentiment, or age-group characteristics). Then PPLM can be seen as modeling the conditional distribution of generated text X given attribute a, i.e., $p(X|a)$. Note that Bayes' theorem ties these three definitions together as follows:

$$p(X|a) \overbrace{=}^{\text{Bayes' theorem}} \frac{p(X)p(a|X)}{p(a)} \propto p(X)p(a|X). \tag{6}$$

To control the generated text, PPLM shifts the aforementioned history H_t (i.e., all Transformer key-value pairs generated up to time t) in the direction of the sum of two gradients:

1. Ascending $\nabla \log p(a|X)$: maximizing the log-likelihood of the desired attribute a under the conditional attribute model. This enforces attribute control.
2. Ascending $\nabla \log p(X)$: maximizing the log-likelihood of the generated language under the original (possibly conversational) language model. This promotes fluency of the generated text.

These two incentive-representing gradients are combined with various coefficients, yielding a set of tunable knobs to steer the generated text in the direction of the desired fluency, attribute control, and length.

Let's first focus on the first of the two gradients, i.e., the attribute control promoting $\nabla \log p(a|X)$. ΔH_t represents the update to history H_t that pushes the distribution of the generated text X in the direction that has a higher likelihood of adhering to desired attribute a. The gradient update rule can be expressed as:

$$\Delta H_t \leftarrow \Delta H_t + \alpha \frac{\nabla_{\Delta H_t} \log p(a|H_t + \Delta H_t)}{\|\nabla_{\Delta H_t} \log p(a|H_t + \Delta H_t)\|^{\gamma}} \tag{7}$$

where α is the step size, and γ denotes the normalization term's scaling coefficient. Both step size (α) and the scaling coefficient (γ) influence attribute control. Attribute control can be softened by either decreasing α or increasing γ and vice versa. Note that $\alpha = 0$ recovers the original uncontrolled underlying language model (e.g., GPT-2 or DialoGPT). In practice, ΔH_t is initialized at zero, and the update rule in Equation 7 is applied m times (usually 3 to 10), resulting in the updated key-value pair history $\tilde{H}_t = H_t + \Delta H_t$. Then the updated history $\tilde{H}_t$ is passed through the language model, yielding the updated logits (final Transformer-layer): $\tilde{o}_{t+1}, H_t = \texttt{LM}(x_t, \tilde{H}_t)$. And finally the shifted $\tilde{o}_{t+1}$ is linearly mapped through a softmax layer to produce a new, more attribute-adherent, distribution from which to sample, i.e., $x_{t+1} \sim \tilde{p}_{t+1} = \texttt{softmax}\left(W \tilde{o}_{t+1}\right)$.

The method described until now will generate attribute-adherent text, but will likely yield fooling examples [Nguyen et al., 2015] that are gibberish to humans, but get assigned high $p(a|x)$ by the attribute model [Dathathri et al., 2020]. That is why Dathathri et al. [2020] apply two methods to ensure fluency of the generate text. The first is to update ΔH_t such to minimize the Kullback-Leibler (KL) divergence (denoted D_{KL}) between the shifted and original distributions. In practice, D_{KL} is scaled by a coefficient λ_{KL}, typically found to work well for most tasks when set to 0.01. Repetitive text generation (i.e., high $p(a|x)$ but low $p(x)$) can therefore sometimes be avoided by increasing λ_{KL}. The second method to ensure fluency is Post-norm Geometric Mean Fusion [Stahlberg et al., 2018] which, instead of directly influencing ΔH_t like minimizing D_{KL}, fuses the altered generative distribution $\tilde{p}_{t+1}$ with the unconditional language distribution $p(x)$. This is done during generation by sampling the next token as follows:

$$x_{t+1} \sim \frac{1}{\beta}\left(\tilde{p}_{t+1}^{\gamma_{gm}} p_{t+1}^{1-\gamma_{gm}}\right) \tag{8}$$

where β is a normalization constant, p_{t+1} and $\tilde{p}_{t+1}$ denote the original and modified distributions, respectively, and γ_{gn} is a scaling term that interpolates between the two distributions. Because the new sampling distribution in Equation 8 converges towards the unconditional language model as $\gamma_{gm} \to 0$, repetitive text generation can be avoided by decreasing the scaling term.

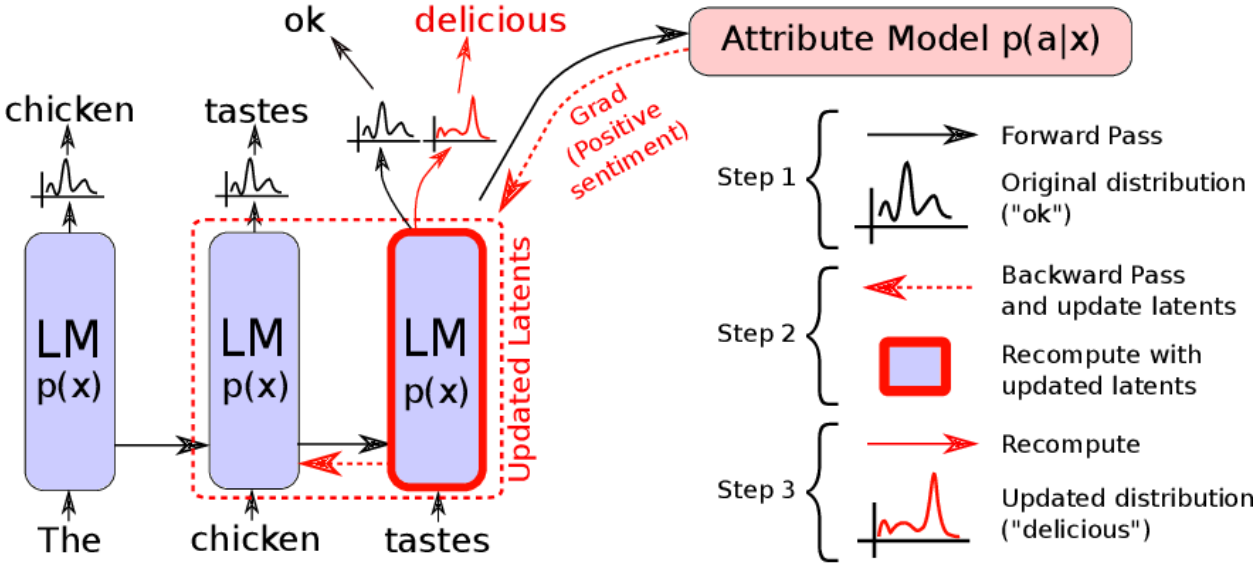

Figure 2: A schematic overview of the plug-and-play interaction between attribute model $p(a|\mathbf{x})$ and language model $p(\mathbf{x})$. *Original image source:* Figure 1 of Dathathri et al. [2020]

It is important to realize that the plug-and-play method applied by Dathathri et al. [2020] and Madotto et al. [2020] is different from fine-tuning. Note that in Equation 7 the gradient updates are restricted to the history H_t, and do not affect the model's parameters. Because the key-value pairs $(K_t^{(i)}, V_t^{(i)})$ that comprise H_t are activations and not model-weights, the updates only take place in the activation-space. This means that PPLM leaves the underlying (conversational) language model untouched.

Contrary to fine-tuning often massive LMs, PPLM does not incur a significant training cost (depending of course on the complexity of the discriminator or attribute model). However, Madotto et al. [2020] show that PPLM needs a fixed number of m update-steps to for every generated token. This makes the original PPLM setup unsuitable for online interactive applications, like conversational systems. Addressing this problem, they introduce plug-and-play conversational models (PPCM), which extends PPLM by using the original model setup to generate dialogue datasets with the desired attribute a, and then use optimized residual adapters [Bapna and Firat, 2019] to control `LM`'s output distribution.

Residual adapters are optimizable modules stacked on every Transformer-layer of a pre-trained (language) model. The adapter module then steers the Transformer's output distribution without changing the pre-trained model's weights. A Layer Normalization module [Ba et al., 2016] followed by an auto-encoder with residual a connection constitutes a residual adapter module. More specifically, the residual adapter block can be expressed as the following function composition:

$$f_{\theta_i}(x) = \texttt{ReLU}(\texttt{LayerNorm}(x) \cdot W_i^E) \cdot W_i^D,$$
$$\texttt{Adapter}(o_{:t}^i) = f_{\theta_i}(o_{:t}^i) + o_{:t}^i \tag{9}$$

where $o_{:t}^i \in \mathbb{R}^{t \times d}$ denotes the Transformer's i-th layer's latent output at step t, d is the hidden state's size, W_i^E and W_i^D are learnable parameter-matrices of sizes $d \times m$ and $m \times d$, respectively. Finally, m is the auto-encoder's bottle-neck dimension, which is a tunable hyper-parameter for changing the residual adapter's capacity. In practice, Madotto et al. [2020] use PPLM to generate n attribute-adherent dialogue datasets $\mathscr{D}^a = \{\mathcal{D}^1, ..., \mathcal{D}^n\}$, for attribute a. These generated dialogue datasets are then used to train the residual adapter, which they aptly name a plug-and-play adapter, so it can be used to control the language model's output distribution. So for every attribute a, they train the plug-and-play adapter's parameters $\Theta_a := \{\theta_0^a, ..., \theta_l^a\}$, where $\theta_i^a := \{W_i^{E,a}, W_i^{D,a}\}$, such that negative log-likelihood over the corresponding dialogue dataset $\mathscr{D}^a$ is minimized:

$$\Theta_a \text{ s.t. } \min \mathcal{L}(\mathscr{D}^a) = -\sum_{k}^{|\mathscr{D}^a|} \sum_{i}^{n} \log p(s_i^k | s_{<i}^k, \mathcal{D}_t^k), \tag{10}$$

where s_i^k is the i-th generated token of response $S_t^k = \{s_0^k, ..., s_n^k\}$ with maximum sequence length n.

5 Planning

Table 1: Timetable for my planning on experimenting, evaluating, and writing.

Month	Week	Task
June-July	26	• 28/06 Thesis proposal deadline. • Pre-summer RPA meeting roundup. • Migrate from AWS EC2 to Lisa GPU. • Construct properly representative most frequently used words per age-group (unique to age-group). • First experiments with GPT-2 + BNC Age discriminator.
July	27	• Break.
July	28	• Break.
July	29	• Coding for BNC-discrim. + GPT-2 • Coding for BNC-discrim. + DialoGPT • Latent space and feature inspections age-discriminators. • PPCM code setup.
July-August	30	• Coding for BAC-discrim + GPT-2 • Coding for BAC-discrim + DialoGPT • Automatic evaluation age-discrim. + LMs. • Write short summaries and interpretations of results.

August	31	• Finalize writing of age-discrim. + LM results up until now. • Break from Friday 06/08 to Monday 09/08.
August	32	• Coding for user-ready adaptive system (PPCM). • Writing for results and discussion until now.
August	33	• Coding for user-ready adaptive system (PPCM). • Writing for results and discussion until now.
August	34	• Test-runs for user-ready adaptive system (PPCM). • Writing for results and discussion until now.
August-September	35	• Test-runs for user-ready adaptive system (PPCM). • Writing for results and discussion until now.
September	36	• First RPA meeting after summer. Set up plans for next phase of experiments.
September	37	• Phase 2 user experiments. • Write tentative results.
September	38	• Phase 2 user experiments. • Write tentative results.
September-October	39	• 01/10 Submit draft of final version for feedback.
October	40	• Time for additional experiments and case studies.
October	41	• Time for additional experiments and case studies.
October	42	• Revise final version.
October	43	• Revise final version.
November	44	• 01/11 Final thesis submission deadline.

References

E. E. Abdallah, J. R. Alzghoul, and M. Alzghool. Age and gender prediction in open domain text. *Procedia Computer Science*, 170:563–570, 2020.

L. J. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016. URL http://arxiv.org/abs/1607.06450.

J. Baan, M. ter Hoeve, M. van der Wees, A. Schuth, and M. de Rijke. Do transformer attention heads provide transparency in abstractive summarization? *CoRR*, abs/1907.00570, 2019. URL http://arxiv.org/abs/1907.00570.

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.0473.

A. Bapna and O. Firat. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1538–1548, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1165. URL https://www.aclweb.org/anthology/D19-1165.

C. M. Bishop. *Pattern recognition and machine learning*. springer, 2006.

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

N. Dai, J. Liang, X. Qiu, and X. Huang. Style transformer: Unpaired text style transfer without disentangled latent representation. *arXiv preprint arXiv:1905.05621*, 2019.

S. Dathathri, A. Madotto, J. Lan, J. Hung, E. Frank, P. Molino, J. Yosinski, and R. Liu. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=H1edEyBKDS.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

C. Gallois and H. Giles. Communication accommodation theory. *The international encyclopedia of language and social interaction*, pages 1–18, 2015.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

N. S. Keskar, B. McCann, L. Varshney, C. Xiong, and R. Socher. CTRL - A Conditional Transformer Language Model for Controllable Generation. *arXiv preprint arXiv:1909.05858*, 2019.

D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

G. Lample, S. Subramanian, E. Smith, L. Denoyer, M. Ranzato, and Y.-L. Boureau. Multiple-attribute text rewriting. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1g2NhC5KQ.

C. Li, X. Gao, Y. Li, B. Peng, X. Li, Y. Zhang, and J. Gao. Optimus: Organizing sentences via pre-trained modeling of a latent space. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4678–4699, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.378. URL https://www.aclweb.org/anthology/2020.emnlp-main.378.

R. Love, C. Dembry, A. Hardie, V. Brezina, and T. McEnery. The spoken bnc2014: designing and building a spoken corpus of everyday conversations. *In International Journal of Corpus Linguistics*, 22(3):319–344, 2017.

A. Madotto, E. Ishii, Z. Lin, S. Dathathri, and P. Fung. Plug-and-play conversational models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2422–2433, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.219. URL https://www.aclweb.org/anthology/2020.findings-emnlp.219.

A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*. IEEE, 2015.

D. Nguyen, N. A. Smith, and C. P. Rosé. Author age prediction from text using linear regression. In *Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 115–123, Portland, OR, USA, June 2011. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W11-1515.

D. Nguyen, D. Trieschnigg, A. S. Doğruöz, R. Gravel, M. Theune, T. Meder, and F. De Jong. Why gender and age prediction from tweets is hard: Lessons from a crowdsourcing experiment. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1950–1961, 2014.

J. W. Pennebaker and L. D. Stone. Words of wisdom: Language use over the life span. *Journal of Personality and Social Psychology*, 85(2):291–301, 2003. URL `https://doi.org/10.1037/0022-3514.85.2.291`.

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

A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

J. Schler, M. Koppel, S. Argamon, and J. W. Pennebaker. Effects of age and gender on blogging. In *AAAI spring symposium: Computational approaches to analyzing weblogs*, volume 6, pages 199–205, 2006.

R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, Aug. 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL `https://www.aclweb.org/anthology/P16-1162`.

F. Stahlberg, J. Cross, and V. Stoyanov. Simple fusion: Return of the language model. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 204–211, Brussels, Belgium, Oct. 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6321. URL `https://www.aclweb.org/anthology/W18-6321`.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

G. Zeng, W. Yang, Z. Ju, Y. Yang, S. Wang, R. Zhang, M. Zhou, J. Zeng, X. Dong, R. Zhang, H. Fang, P. Zhu, S. Chen, and P. Xie. MedDialog: Large-scale medical dialogue datasets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9241–9250, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.743. URL `https://www.aclweb.org/anthology/2020.emnlp-main.743`.

Y. Zhang, S. Sun, M. Galley, Y.-C. Chen, C. Brockett, X. Gao, J. Gao, J. Liu, and B. Dolan. Dialogpt: Large-scale generative pre-training for conversational response generation. In *ACL, system demonstration*, 2020.

Y. Zheng, G. Chen, M. Huang, S. Liu, and X. Zhu. Personalized dialogue generation with diversified traits. *arXiv preprint arXiv:1901.09672*, 2019.