University of Amsterdam

MSc Artificial Intelligence
Master Thesis

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

by

Lennert Jansen

10488952

October 18, 2021

Number of Credits
Period in which the research was carried out

Supervisor:
Dr Sandro Pezzelle
Dr Raquel Fernández
Dr Arabella Sinclair

Assessor:
???

Institute for Logic, Language and Computation

Contents

Chapter 1

Introduction

In recent years, we have witnessed promising advances in 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. 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.

[L: CTG comes out of the blue in the following paragraph. Introduce it a little bit by describing what is is, and why/how it is an important task.]

- Controllable text generation entails generating text samples that possess a predefined textual property, like having a positive sentiment, or being about a certain topic.
- Controlling more fine-grained linguistic properties, like resemblance of age-specific vernacular, still poses an important, yet unsolved/insufficiently studied (?) challenge.
- Personalized interaction between humans and AI systems is crucial to obtain systems that can be trusted by users and are perceived as natural.
- (Age-)adaptive language generation can be used to personalize AI-powered personal assistants like Siri and Alexa, improving user experience and trust.

- It is important for AI-power conversational agents to be accessible to varying user profiles, rather than targeted at one particular user group.
- In this work, I/we focus on one aspect that may influence successful personalization of conversational agents: user age profile.

Controllable text generation (CTG) aims 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.

[L: Introduce dialogue response generation a bit more. Also emphasize its importance. And then introduce the combined task and its importance.] A blend of CTG and dialogue response generation, 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, gender, or other persona-labeled traits via language expression [Zheng et al., 2019].

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, 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 thesis is structured as follows: Chapter 2.1 positions the subject of controlled text generation in its theoretical background, and and 2.2 compares it to the most relevant related work. The methodology in Chapter **??** gives detailed explanations of the most important modeling methods and techniques used for this research.

- When introducing your own work and proposing your hypothesis, use the following argument: *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.*

Chapter 2

Literature review

This chapter is a two-part literature review. The first section, Background or Section 2.1, provides an overview of this thesis' central problem of controllable dialogue generation, and the components involved, i.e., dialogue, (controllable) language generation, dialogue response generation, and age modeling. In the second section, Related work or Section 2.2, I discuss previous approaches, relevant to my work, that have been proposed to tackle each of these components, either separately or jointly. Approaches to different, but strongly related, problems, like text style transfer, are also described in Section 2.2.

2.1 Background

[L: Keep in mind the following distinction between Background and Related Work - The Background section should give an overview of the problem and the components involved: dialogue, language generation, dialogue response generation, age modelling, etc., without focusing on one or the other approach — in Related Work, you describe approaches that have been proposed to tackle each of these components, separately or jointly, and which are related or relevant to your own work for some reason]

We focus on controllable language generation, i.e., endowing automatically produced text with certain desired linguistic characteristics, and apply it to dialogue. This naturally involves having a model for language generation that can be controlled to write texts passages with different linguistic styles. In what follows, we introduce the crucial concepts behind this: models for language generation, methods to control the output, dialogue, and age modeling.

[L: isn't this redundant, given the chapter's introduction?]

2.1.1 Language Models

Generally speaking, language modeling is central to many NLP tasks. A language model (LM) is a probability distribution over words in a sentence or document. Language models are trained to predict the probability of the next word in an sentence, given the preceding sequence of words. The language modeling 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 factorized as the product of its constituent conditionals [Radford et al., 2019]:

$$p(\mathbf{x}) = \prod_{j=1}^{n} p(s_j | s_1, ..., s_{j-1}). \tag{2.1}$$

This formulation 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 as 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 modeling 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].

2.1.2 (Controllable) Text Generation

In text generation, a language model $p(\mathbf{x})$ is asked to produce text $\mathbf{x}$ given a prompt by sampling 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 represents a linguistic characteristic of the text, 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 it is intelligible for a machine is a challenge in itself. Second, like many NLP problems, there are not many parallel corpora. In the context of controllable generation, parallel corpora are datasets of target and source texts that only differ with respect to some attribute. 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.

2.1.3 Dialogue

Dialogue can be described as a written or spoken conversational exchange between two or more interlocutors. Generally speaking, its purpose is to exchange information or build relationships between interlocutors. A dialogue typically consists of interlocutors exchanging utterances in turns.

[L: **TODO** - Give an example of a dialogue snippet (from the BNC)]

Dialogue distinguishes itself from discourse in that it necessarily involves two or more participants exchanging information and contributing to the conversation, whereas discourse can be a one-way exchange of information, like a lecture or blog-post.

The NLP operationalization of dialogue, dialogue generation, is a fundamental component for real-world virtual assistants such as Siri and Alexa. It is the task of automatically generate a response given a prompt by the user.

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.

2.1.4 Dialogue response generation

Text generation is suitable to tackle tasks such as 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, while remaining relevant to the prompt.

2.1.5 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. Assuming an encoder-decoder setup, most personalized neural conversation models can be classified as one of two types: implicit and explicit personalisation models [Zheng et al., 2019]. 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.

When generating responses, *explicit* personalization models are conditioned either on a given personal profile, text-described persona, or simply an attribute label. That is, 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. 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.

2.1.6 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, Zheng et al., 2019]. 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*). Pennebaker and Stone [2003] report 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].

2.2 Related work

[L: Keep in mind the following distinction between Background and Related Work - The Background section should give an overview of the problem and the components involved: dialogue, language generation, dialogue response generation, age modeling, etc., without focusing on one or the other approach — in Related Work, you describe approaches that have been proposed to tackle each of these components, separately or jointly, and which are related or relevant to your own work for some reason]

2.2.1 Automated age detection

[L: Consider adding a small sub-section about automated age detection from text, because you often bring up the problem and other researchers' approaches to solving it in the Background section.]

...

[L: work this into this subsection. Taken from workshop paper submission]

Previous work on age detection in dialogue has focused on speech features, which are known to systematically vary across age groups. For example, Wolters et al. [2009] learn logistic regression age classifiers from a small dialogue dataset using different acoustic cues supplemented with a small set of hand-crafted lexical features, while Li et al. [2013] develop SVM classifiers using acoustic and prosodic features extracted from scripted utterances spoken by participants interacting with an artificial system. In contrast to this line of work, we investigate whether different age groups can be detected from textual linguistic information rather than voice-related cues. We explore whether, and to what extent, various state-of-the-art NLP models are able to capture such differences in dialogue data as a preliminary step to age-group adaptation by conversational agents. We build on the work of Schler et al. [2006], who focus on age detection in written discourse using a corpus of blog posts. The authors learn a Multi-Class Real Winnow classifier leveraging a set of pre-determined style- and content-based features, including part-of-speech categories, function words, and the 1000 unigrams with the highest information gain in the training set. They find that content features (lexical unigrams) yield higher accuracy (∼ 74%) than style features (∼72%), while their best results are obtained with their combination (∼76%). We extend this investigation in several key ways: (1) we leverage state-of-the-art NLP models that allow us to learn representations end-to-end, without the need to specify concrete features in advance; (2) we apply this approach to dialogue data, using a large-scale dataset of transcribed, spontaneous open-domain dialogues; (3) we show that text-based models can indeed detect age-related differences in both discourse and dialogue, even in the case of very sparse *signal at the level of dialogue utterances; and finally (4) we carry out an in-depth analysis of the models' predictions to gain insight on which elements of language use are most informative.*

...

More recent studies, like that of Nguyen et al. [2011], Zheng et al. [2019], 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.

2.2.2 Controllable language generation

Previous approaches to controlled language generation require fine-tuning large Transformer-based 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 require fine-tuning for control, like CTRL, can produce high quality fluent text because they are specifically trained to maximize the likelihood of generated sequences, given an attribute (denoted $p(\mathbf{x}|a)$), but require training massive language models with computational costs.

Other 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 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., 2019]. Nevertheless, much like the previously mentioned CTG models, OPTIMUS still incurs a significant computational cost for fine-tuning per NLP task.

[L: TODO: Where does the following sentence fit best? "The plug-and-play setup of PPLM forms one of the main theoretical foundations of this work."]

The plug-and-play language model (PPLM) [Dathathri et al., 2020] is a recent solution to the problem of high re-training costs of controlled language generation. This approach, inspired by a similar technique for style-control of generated images [Nguyen et al., 2017], leverages the fluency of large-scale language models when controlling them for a specific linguistic attribute, while avoiding incurring significant costs of fine-tuning these massive language models. The main benefit of this setup is its low-cost extensibility. Namely, such large-scale language models are often open-source and available online, and can now be tailored to users' specific needs using a significantly easier to train attribute model. The original architecture proposed by Dathathri et al. uses GPT-2 as a 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 of 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 that PPLM achieves desirable fluency (i.e., perplexity measured with GPT(-1) [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.

2.2.3 Text style transfer

Text style transfer is the task of changing a text's stylistic properties, while retaining its style-independent properties, like content and fluency [Dai et al., 2019]. Text style transfer is a closely related problem to controllable language generation. Its similarity lies in trying to modify the output distribution of a language generation model, such that stylistic characteristics of the produced text are controllable, keeping content and fluency preserved. It involves rewriting an input text with a specific style. 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 text style transfer is to produce output text $\hat{\mathbf{x}}$ with style $\hat{\mathbf{s}}$, and the style-independent properties of $\mathbf{x}$.

Previous approaches to text style transfer involve passing input text through an RNN-based encoder, yielding a style-dependent latent representation $\mathbf{z}$ [Zhang et al., 2018]. Typically, these approaches then attempt 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)** It is hard to capture rich semantic information in the latent representation due to limited capacity of vector representations (especially for long texts). **(3)** 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. **(4)** Since most previous approaches use RNN-based encoder-decoder frameworks, they have problems capturing long-range dependencies in the input sentences. Furthermore, disentanglement might be unnecessary, as Lample et al. [2019] have shown a proper decoder can perform controllable text generation from an entangled latent representation by "overwriting" the original style.

To address these drawbacks, Dai et al. [2019] propose Style Transformer, a Transformer-based alternative encoder-decoder framework 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, Dai et al. [2019] contribute a novel training algorithm for such models, that boasts significant improvements of results on two text style transfer datasets.

2.2.4 Dialogue Generation

Dialogue generation is task of automatically generating a response given a user's prompt. 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 highly frequent hypotheses, 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 small (117M), medium (345M), and large (762M) parameter sizes. 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`.

Dialogue generation is the essential precursor to this thesis' ultimate task of controllable dialogue generation.

[L: The previous sentence feels out of the blue. Consider removing it or think of a way to create a natural flow towards it.]

2.2.5 Controlled Dialogue Generation

Controlled dialogue generation is the task of steering automatically generated conversational responses to possess desired attributes, like sentiment, topic, or more abstract writing style characteristics. 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 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. They introduce a dataset `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. Zheng et al. [2019] also propose persona-aware models that include 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 (such as LSTMs or gated recurrent units (GRUs)), convolutional neural networks (CNNs), or hybrids of these systems (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 are able to produce human-like conversational responses, sometimes even leveraging the fluency of large pre-trained LMs, they all suffer from the same computational drawback. They all require massive amounts of computational power to adapt their language styles, because in their 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. Due to the computational complexity of PPLM's decoding process, PPLM 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 computational resources. See Section 4.2.4 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). Previous work on controllable language generation focuses on content (e.g., topical attributes, or sentiment), rather than more abstract linguistic features, which I hypothesize are more challenging to model and control. The previously mentioned work by Zheng et al. [2019] is a notable exception, as their approach deals with controlling dialogue systems for linguistic features, like age, gender, and geographical region. However, Zheng et al. [2019] still suffers from significant computational costs, because control is achieved by fine-tuning a large system for every specific set of attributes. Furthermore, their proposed architectures are RNN-based, as opposed to my Transformer-based approach. My work therefore aims to extend the applicability of plug-and-play controlled generation to more abstract linguistic characteristics than those explored by Dathathri et al. [2020] and Madotto et al. [2020], and without the significant fine-tuning cost of Zheng et al. [2019].

[L: TODO

- Ask for Sandro's feedback on this last rephrased paragraph.
- Is it worth mentioning that Zheng et al 2019 deals with Chinese Mandarin dialogue systems, and mine with English?

]

Chapter 3

Experiment 1: Classification

3.1 Introduction

This chapter focuses on our experiments about age detection from text, and the components involved. The problem we tackle in this first phase of experiments is automated detection of age-related linguistic patterns from dialogue and discourse, using current text-based NLP models. Being able to detect and investigate these linguistic differences is important for controlled dialogue generation, because it suggests that adapting automated conversational responses to a user's age is possible. Moreover, it can provide us with insights about which linguistic features are most salient for distinguishing between, and adapting to, different age groups. We expect that the classification models are able to reliably detect age-related differences in both transcribed dialogue and discourse, and the most informative differences to lie at the syntactic-level.

The following section describes the two datasets used for these experiments. There we provide descriptive statistics, examples, and comparisons between the corpora. Section 3.3 covers the problem description in more detail, along with the models used, and our experimental setup. The classification results are presented in Section 3.4. Then for the dialogue classification models, Section 3.5 contains both quantitative and qualitative analyses of the results.

3.2 Data

We use a dataset of dialogue data where information about the age of the speakers involved in the conversation is available (see the dialogue snippets in Figure), i.e., the spoken partition of the British National Corpus Love et al. [2017]. We henceforth refer to it as our *dialogue* dataset. For

age 19-29
A: oh that's cool
B: different sights and stuff
A: oh

age 50+
A: well quite and I'd have to come back as well
B: that's of course
A: and make up for you know

Figure 3.1: Example dialogue snippets from speakers of different age groups (19-29 vs. 50+) in the British National Corpus. We conjecture that stylistic and lexical differences between age groups can be detected. In our approach, we experiment at the level of the utterance.

comparison with previous work, and to explore commonalities and differences between various types of language data, we also experiment with a dataset of discourse, i.e., the Blog Authorship Corpus used by Schler et al. [2006], that we henceforth refer to as our *discourse* dataset. Below, we briefly describe the two datasets along with the pre-processing steps we took to make the data suitable for our experiments.

dataset	**# age groups**	**# samples**	**# tokens**	**mean length (±std)**	**min - max length**	**# topics**
dialogue	2	67,282	787,352	11.7 (±19.0)	1 - 1246	790
discourse	3	677,244	140M	102.2 (±212.9)	1 - 71,580	40

Table 3.1: Descriptive statistics of the datasets used in our experiments. Length is the number of tokens in a sample.

3.2.1 Dialogue Dataset

This partition of the British National Corpus includes spoken informal open-domain conversations between people that were collected between 2012 and 2016 via crowd-sourcing, and then recorded and transcribed by the creators. Dialogues can be between two or more interlocutors, and are annotated along several dimensions including age and gender together with geographic and social indicators. Speaker ages in the original dataset are categorized in the following ten brackets: 0-10, 11-18, 19-29, 30-39, 40-49, 50-59, 60-69, 70-79, 80-89, and 90-99.

We focus on conversations in the British National Corpus that took place between two interlocutors, and only consider dialogues between people of the same age group. We then focus on dialogues by speakers belonging to two age groups: 19-29 and 50+, in which we group conversations from five original brackets: 50-59, 60-69, 70-79, 80-89, and 90-99. We omit the intermediate age bracket to allow for clearer differentiation.

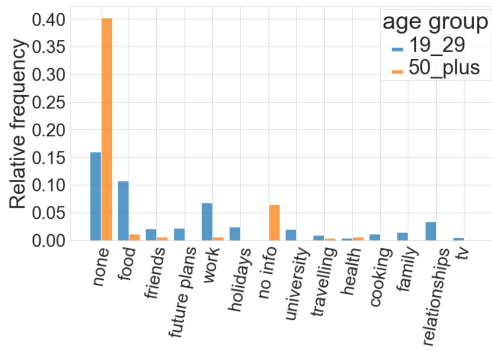
We split the dialogues into their constituent utterances (e.g., from each dialogue snippet in Figure 3.1 we extract three utterances), and further pre-process them by removing non-alphabetical characters. Only samples which were not empty after pre-processing were kept. The resulting dialogue dataset, that we use for our experiments, includes around 67K utterances with an average length of 11.7 tokens. Descriptive statistics of it are reported in Table 3.2.

Each conversation in the British National Corpus is annotated with a list of *topics* provided by the speakers during data collection. To extract a single representative topic from this list, we first compute the frequency of all topic labels in the whole dataset. Then, for each utterance, we take the label in the conversation with the highest frequency in the ranking. In total, our final dataset includes 790 unique topic labels. The distribution of the most frequent ones is reported in Figure 3.2a. As can be seen, frequent topics (besides the frequent *none* label) are *food*, *work*, and *holidays*, which reveals the colloquial and everyday nature of the dialogues in this dataset.

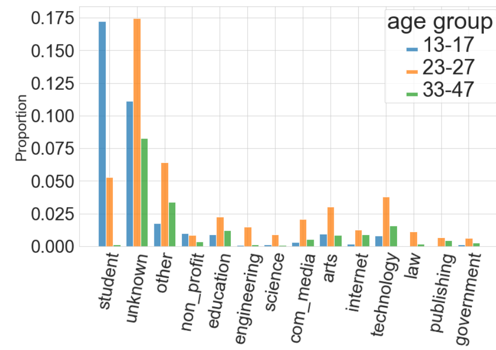
3.2.2 Discourse Dataset

The Blog Authorship Corpus Schler et al. [2006] is a collection of blog posts posted on `https://www.blogger.com`, gathered in or before August 2004. Each blog entry is written by a single user whose age, gender, and astrological sign are reported. The corpus contains almost 700,000 posts by 19,000 unique bloggers (i.e., ∼35 posts per blogger on average). For our experiments, similar to Schler et al. [2006], we consider three age groups: 13-17, 23-27, and 33+. We pre-process the data in the same way as described above, namely by removing stopwords and non-alphabetical characters. The resulting dataset, that we use for our experiments, includes slightly more than 677K samples with an average length of 102.2 tokens. Descriptive statistics of it are reported in Table 3.2. [L: TODO - Adjust paragraph to W.S. setting.]

Each sample in the Blog Authorship Corpus is annotated with one topic. In our final discourse dataset, the unique topics present are 40. Figure 3.2b reports the distribution of the most frequent ones. As can be noted, frequent topics are *student*, *arts*, and *technology*, which reveals that this and the dialogue dataset are rather different.



(a) Distribution of most frequent topics (including the *none* and *no info* labels) in the **dialogue dataset**, shown by age group. Best viewed in color.



(b) Distribution of most frequent topics (including the *unknown* label) in the **discourse dataset**, shown by age group. Best viewed in color.

Figure 3.2

3.3 Methodology and experimental setup

The current section contains the methodology and experimental details of our automated age-detection experiments. We frame the problem as a N-class classification problem: given a fragment of text X, we seek to predict the age class of its speaker/writer. For the dialogue dataset, $N = 2$, while $N = 3$ for the discourse dataset. We experiment with various models, that we briefly describe here below. Details on the training and evaluation of models are given at the end of the sub-section.

***n*-gram** Our simplest models are based on n-grams, which have the advantage of being highly interpretable. Each data entry (i.e., a dialogue utterance or blog post) is split into chunks of all possible contiguous sequences of n tokens. The resulting vectorized features are used by a logistic regression model to estimate the odds of a text sample belonging to a certain age group. We experiment with unigram, bigram and trigram models. Note that a bigram model uses unigrams and bigrams, and a trigram model unigrams, bigrams, and trigrams.

LSTM and BiLSTM We use a standard Long Short-Term Memory network [LSTM; Hochreiter and Schmidhuber, 1997] with two layers, embedding size 512, and hidden layer size 1024. Batch-wise padding is applied to variable length sequences. The original model's bidirectional extension, the bidirectional LSTM [BiLSTM; Schuster and Paliwal, 1997], is also used. BiLSTM more thoroughly leverages forward and backward directed information by combining the hidden states from both directions. Padding is similarly applied to this model, and the following optimal architecture is found: embedding size 64, 2 layers, and hidden layer size 512. Both RNN-model are found to perform optimally for a learning rate of 10^{-3}.

BERT We experiment with a Transformer-based model, i.e., Bidirectional Encoder Representations from Transformers [BERT; Devlin et al., 2019] for text classification. BERT is pre-trained to learn deeply bidirectional language representations from massive amounts of unlabeled textual data. We experiment with the base, uncased version of BERT, in two settings: by using its pre-trained frozen embeddings (BERT$_{frozen}$) and by fine-tuning the embeddings on our age classification task (BERT$_{FT}$). The BERT embeddings are followed by a dropout layer with dropout probability 0.1, and a linear layer with input size 768.

Experimental details Both datasets are randomly split into a training (75%), validation (15%), and test (10%) set. Each model with a given configuration of hyperparameters is run 5 times with different random initializations. All models are trained on an NVIDIA TitanRTX GPU.

The n-gram models are trained in a One-vs-Rest (OvR) fashion, and optimized using the Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm Liu and Nocedal [1989], with a maximum of 10^6 iterations. The n-gram models are trained until convergence or for the maximum number of iterations.

LSTMs and BERT-based models are optimized using Adam [Kingma and Ba, 2015], and trained for 10 epochs, with an early stopping patience of 3 epochs. The RNN-based models' embeddings are jointly trained, and optimal hyperparameters (i.e., learning rate, embedding size, hidden layer size, and number of layers) are determined using the validation set and a guided grid-search. BERT$_{FT}$ is fine-tuned on the validation set for 10 epochs, or until the early stopping criterion is met. BERT models have a maximum input length of 512 tokens. Sequences exceeding this length are truncated.

3.4 Detecting Age-Related Linguistic Patterns in Dialogue

We first report results on *discourse* to check whether we replicate previous findings. Then, we focus on *dialogue* to answer our research questions. We report accuracy and F_1 for each age group.

3.4.1 Classification performance on discourse

Table 3.2 reports the results. As can be seen, all models are well above the baseline in terms of both accuracy and F_1s. This overall confirms previous evidence Schler et al. [2006] that language features of (written) *discourse* can predict, to some extent, the age group to which the person belongs. At the same time, BERT fine-tuned on the age classification task stands out as our best-performing model by achieving highest accuracy (0.731) and highest F_1 in all age groups. BiLSTM and LSTM rank second (0.720) and third (0.714) in terms of accuracy, respectively, while a somehow more mixed pattern is observed for F_1 scores.

Overall, these results indicate that powerful neural models that are capable of representing the linguistic context have a great advantage on this dataset over simpler n-gram models, which are more than 10 accuracy points behind.

Finally, it should be noted that our best results are slightly lower than those obtained by Schler et al. [2006]. This could be due to two main reasons: First, they experiment with a different (smaller) dataset than ours,[1] which also has a different majority baseline (see Table 3.2). Second, while in our approach all models are trained end-to-end on the task, Schler et al. [2006] use hand-crafted features that are specific to the dataset, (as mentioned in the Introduction), which could constitute an advantage.

3.4.2 Classification performance on dialogue

Table 3.3 reports the results. As can be seen, BERT fine-tuned on the task is again the best-performing model in terms of accuracy (0.729), which confirms the effectiveness of this model in detecting age-related linguistic differences. At the same time, it can be noted that the model based on trigrams is basically on par with it in terms of accuracy (0.722) and well above both LSTM and BiLSTM (0.693 and 0.691, respectively). A similar pattern is shown for F_1 scores, where BERT fine-tuned and the trigram model achieve comparable performance, with LSTMs being overall behind.

Overall, our results indicate that predicting the age group to which a speaker belongs, using text-based models, is possible also for *dialogue* data, though the task appears to be somehow more challenging compared to when performed on discourse (note that the improvement with respect to the majority/random baseline is lower in dialogue). At the same time, the different ranking of models observed between discourse and dialogue suggests possibly different strategies

[1] They are left with roughly 511K datapoints after pre-processing, while we experiment with around 677K.

used by models to solve the task. In particular, the very good performance of the trigram model in *dialogue* suggests that leveraging 'local' linguistic features captured by n-grams is extremely effective in this setup. This could indicate that differences among various age groups are at the level of local lexical constructions. This deserves further analysis, which we carry out in the next section.

Model	**Accuracy** ↑ better	$F_1^{(13-17)}$ ↑ better	$F_1^{(23-27)}$ ↑ better	$F_1^{(33+)}$ ↑ better
Majority class	0.472	*	0.642	*
Schler et al. [2006]	0.762	0.860	0.748	0.504
unigram	0.603 (0.001)	0.760 (0.003)	0.706 (0.001)	0.491 (0.003)
bigram	0.627 (0.001)	0.788 (0.001)	0.715 (0.001)	0.504 (0.002)
trigram	0.625 (0.002)	0.789 (0.001)	0.716 (0.002)	0.485 (0.003)
LSTM	0.714 (0.005)	0.772 (0.007)	0.740 (0.004)	0.501 (0.006)
BiLSTM	0.720 (0.001)	0.778 (0.005)	0.746 (0.001)	0.486 (0.016)
BERT_{frozen}	0.604 (0.001)	0.627 (0.011)	0.666 (0.005)	0.198 (0.018)
BERT_{FT}	**0.731** (0.002)	**0.791** (0.003)	**0.752** (0.005)	**0.521** (0.020)

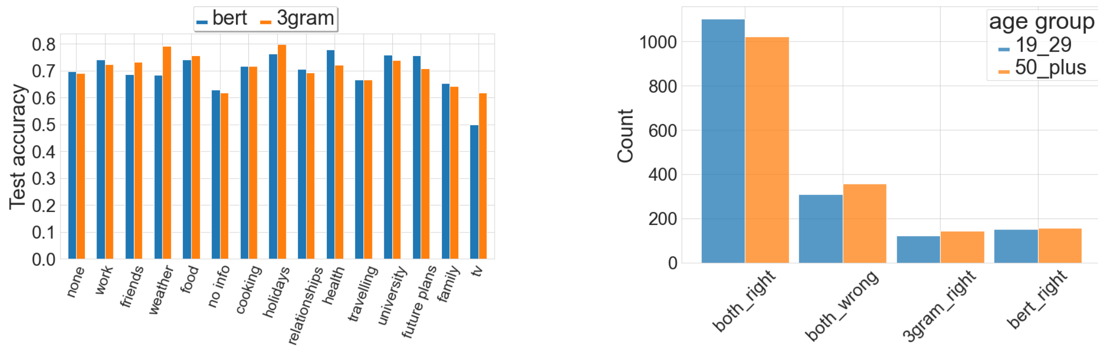
Table 3.2: Discourse dataset. Test set results averaged over 5 random initializations. Format: *average metric (standard error)*. Values in **bold** are the highest in the column; in blue, the second highest. *: F_1 is actually $0/0$.

Model	**Accuracy** ↑ better	$F_1^{(19-29)}$ ↑ better	$F_1^{(50+)}$ ↑ better
Random	0.500	0.500	0.500
unigram	0.701 (0.007)	0.708 (0.009)	0.693 (0.004)
bigram	0.719 (0.002)	0.724 (0.003)	0.714 (0.003)
trigram	0.722 (0.001)	0.727 (0.003)	0.717 (0.001)
LSTM	0.693 (0.003)	0.696 (0.005)	0.691 (0.007)
BiLSTM	0.691 (0.009)	0.702 (0.017)	0.679 (0.007)
BERT_{frozen}	0.675 (0.003)	0.677 (0.008)	0.673 (0.010)
BERT_{FT}	**0.729** (0.002)	**0.730** (0.011)	**0.727** (0.010)

Table 3.3: Dialogue dataset. Test set results averaged over 5 random initializations. Format: *average metric (standard error)*. Values in **bold** are the highest in the column; in blue, the second highest.

age	**both correct**	**both wrong**	**BERT_{FT} correct | trigram wrong**	**trigram correct | BERT_{FT} wrong**
19-29	oh that's cool	A retrospective exhibition	what even on the green slope?	really?
19-29	a text and then I'll do it	chuck them in those pots	yeah you told me to do you told me to do	and she like won't eat any carbs and she's like
19-29	yeah	mm	somebody made the f***ing table	do you not like total greens?
50+	I said no I don't have them	yeah	really?	my under stairs in the kitchen
50+	that's of course	no no that's alright	it's still we we frequently walk that way	in the first place
50+	oh right	what a tragic life	since this this was new this house?	thank you very much

Table 3.4: Examples where both models are correct/wrong or only BERT_{FT}/trigram is correct.



(a) BERT$_{FT}$ and trigram test accuracies per topic for most frequent topics (including none/no info).

(b) Distribution of predicted cases by trigram and BERT$_{FT}$ models for dialogue, split by age groups.

Figure 3.3

3.5 Age detection analyses

We focus our analysis on dialogue. In particular, we compare the two best-performing models, namely BERT$_{FT}$ and the one using trigrams, and aim to shed light on what cues they use to solve the task. We first analyze how these models perform with respect to utterances of various topics. Secondly, we compare the prediction patterns of the two models, which allows us to highlight easy and hard examples. Finally, we focus on the trigram model and report the n-grams that turn out to be most informative to distinguish between age groups.

3.5.1 Performance Against Topic

As described in Section 3.2.1, each utterance in the dialogue dataset is annotated with one label which is representative of its topic.[2] This information is not explicitly available to the models. To explore how the two models deal with utterances in different topical contexts, we compare the accuracy they achieve on the 15 most frequent topics. The results are shown in Figure 3.3a. Two main observations can be made: Firstly, some topics turn out to be generally easier/harder than others, i.e., both models achieve higher/lower performance. To illustrate, both models achieve an accuracy well above 70% on topics like *food*, *holidays* or *university*, while topics such as *tv*, *family* or *no info* appear to be generally more challenging for both models. While this could be due to (a combination of) various factors, one intuitive possibility is that certain topics allow for more discriminative language features, which could be at the level of the lexicon or the style used to talk about them.

[2] In particular, it represents one of the utterance's topic, i.e., the one most frequently used in the whole data.

Secondly, some topics appear to be easier for one model rather than the other, and *vice versa*. To illustrate, the trigram model outperforms BERT on the topics *weather*, *holidays* and *tv*, while an opposite pattern is observed for *work*, *health*, and *future plans*. We conjecture that these patterns could be indicative of different strategies and cues exploited by various models to make a prediction. We explore this issue more in-depth in the following section, where we compare the predictions by the two models and qualitatively inspect some examples.

3.5.2 Comparing Model Predictions

We split the data for analysis by whether or not both models make the same correct or incorrect prediction, or whether they differ. Table 3.6 shows the breakdown of these results. As can be seen, a quite large fraction of samples are correctly classified by both models (63.17%), while in 19.78% cases neither of the models make a correct prediction. The remaining cases are comparably split between cases where only one of the two is correct, with BERT slightly outperforming Trigram by 1.23 percentage points. As shown in Figure 3.3b, the 19-29 age group appears to be be slightly easier compared to the 50+ group, where models are observed to make more errors: the trigram misclassifies 50+ utterances 1.12 times as often as 19-29 utterances, and 1.17 times as often by BERT$_{FT}$.

To qualitatively inspect what the utterances falling into these classes look like, in Table 3.4 we show a few cherry-picked cases for each age group. We notice that, not surprisingly, both models have trouble with backchanneling utterances consisting of a single word, such as *yeah*, *mm*, or *really?*, which are used by both age groups. For example, both models seem to consider *yeah* as a 'young' cue, which leads to wrong predictions when *yeah* is used by a speaker in the 50+ group. As for the utterance *really?*, BERT$_{FT}$ assigns it to the 50+ group, while the trigram model makes the opposite prediction. This indicates that certain utterances simply do not contain sufficient distinguishing information, and model predictions that are based on them should therefore not be considered reliable.

This seems to be particularly the case for short utterances. Indeed, through comparing the average length of the utterances incorrectly classified by both models (rightmost column of Table 3.6), we notice that they are much shorter than those belonging to the other cases. This is interesting, and indicates a key challenge in the analysis of dialogue data: on average, shorter utterances contain less signal. On the other hand, short utterances can provide rich conversational signal in dialogue; for example, backchanneling, exclamations, or other acknowledging acts. As a consequence, using length alone as a filter is not an appropriate approach, as it can remove aspects of language use key to differentiating speaker groups.

3.5.3 Most Informative N-grams

	19-29		50+
coef.	**n-gram**	**coef.**	**n-gram**
-3.20	um	2.37	yes
-2.84	cool	2.12	you know
-2.58	s**t	2.09	wonderful
-2.12	hmm	1.90	how weird
-2.09	like	1.84	chinese
-2.02	was like	1.73	right
-1.96	love	1.71	building
-1.96	as well	1.66	right right
-1.88	as in	1.55	so erm
-1.84	cute	1.43	mm mm
-1.82	uni	1.41	cheers
-1.79	massive	1.39	shed
-1.79	wanna	1.37	pain
-1.79	f**k	1.36	we know
-1.72	tut	1.08	yeah exactly

Table 3.5: For each age group, top 15 most informative n-grams used by the trigram model. **coef.** is the coefficient (and sign) of the corresponding n-gram for the logistic regression model: the higher its absolute value, the higher the utterance's odds to belong to one age group. * indicates masking of foul language.

Analyzing the most informative n-grams used by the trigram model allows us to qualitatively compare the linguistic differences inherent to each age group. In Table A.1 we report the top 15 n-grams per group. We find, firstly and intuitively, that colloquial language seems somewhat generational, with unigrams particularly indicative of younger speakers consisting of words such as *cool* and *massive*, and for older speakers, words like *wonderful*. These unigrams are both informative to the model and indicative of differences in both formality and 'slang' use across age groups.

These most informative n-grams also indicate differences in back-channeling use between age groups; younger speaker's language is more characterized by the use of *hmm*, *um*, *yeah course*, while the top n-grams in the older category will more likely use *yes*, *right*, *right right*. A feature of younger language also apparent from these examples is in their use of more informal language: *yeah course* rather than *yes*. This informal language use also extends to the use of foul language, which make up a percent of the most informative unigrams shown in Table A.1.

Interestingly, while topic words make up many of the most informative n-grams for older speakers in Table A.1, younger speakers are more defined by their use of slang words such as *wanna*, foul language, or adjectives such as *cute*, *cool*, and *massive*. A key finding from Schler et al. [2006] is in the sentiment of language playing an important role, something which some of the most informative n-grams suggest may also be true for the dialogue dataset. As Table A.1 demonstrates, younger speakers use more dramatic language such as negative foul words, and positive *love, cute, cool*; all words with a strong connotative meaning. This prompts us to hypothesize that further inspection is needed to determine whether the same sentiment pattern will be true of dialogue as it has been reported to be in discourse.

	% cases	**avg. length (±std)***
both correct	63.17%	13.51 (±18.98)
both wrong	19.78%	5.82 (±8.33)
only Trigram correct	7.91 %	10.44 (±11.66)
only BERT correct	9.14 %	11.53 (±12.12)

Table 3.6: Percentage (% cases) of (non-)overlapping (in)correctly predicted cases between trigram and BERT$_{FT}$. *Utterance length measured in tokens.

Chapter 4

Experiment 2: Generation

4.1 Introduction

In the previous chapter, we have shown that it is possible to classify younger versus older age groups based on linguistic features. We now aim to check whether it is possible to generate language that encompasses these features. Experiment 1's classifier, which evaluates the generated output of Experiment 2's models, is trained on the spoken component of the BNC, which is a dialogue dataset. Now, in generating, it is important that we generate something similar to a dialogue turn, i.e., a response to a dialogue "prompt".

We will use state-of-art models, GPT-2 [Radford et al., 2019] and DialoGPT [Zhang et al., 2020], for (controlled) language generation. The deliberate choice to use BERT-based models for classification (and evaluation of generated output), and GPT-based models for generation is motivated by the following reasons. BERT's encoder-based bidirectional architecture makes it more suitable for sequence classification than for generation [Devlin et al., 2019]. By similar reasoning, GPT's decoder-based structure makes it a more suitable choice for generation tasks. Furthermore, Experiment 1's best classifier is a fully fine-tuned BERT model (ca 110M parameters), while fine-tuning GPT-2-medium (ca 345M parameters) is infeasible with my computational resources. Finally, using a separate model class (i.e., BERT) for evaluation of GPT-based generation models makes the results more generalizable.

This chapter covers the methodology, experimental setup, results, and analyses relating to the language generation experiments of this thesis. The central problem of this chapter is a plug-and-play approach to age-adaptive dialogue generation. In other words, I seek to use large pre-trained language models for controllable dialogue generation, using activation-space perturbations instead of fine-tuning. Either GPT-2-medium or DialoGPT-medium is used as a base language model. And adaptation of the generated language to a certain age group is achieved using either a linear discriminator or a bag-of-words (BoW), trained or empirically constructed from the dialogue dataset [Love et al., 2017]. Using GPT-2 and DialoGPT as baselines, a set of automated and human evaluation metrics are used to evaluate the fluency and control of the proposed models. I expect discriminator-based control to be more detectable and fine-grained than BoW-based control. I also expect BoW-based control to take a smaller toll on fluency.

The rest of this chapter is structured as follows. Section 4.2 contains detailed descriptions of the most important methods and models involved: Transformers, conversational language modeling, and plug-and-play language models. This section ends with the experimental details, and an explanation of the chosen evaluation metrics. The results of the language generation experiments are presented and interpreted in Section 4.3, followed by the outcomes of various quantitative and qualitative analyses in Section 4.4. A separate section about data is omitted in this chapter, because we use the previously mentioned dialogue dataset for these experiments. The reader is directed to Sub-section 3.2.1 for a detailed description of that corpus, and our pre-processing steps.

4.2 Methods for controlled language generation

Plug-and-play language generation entails using a attribute model to make activation-space perturbations on the output of a pre-trained language model. I explain the important architectures involved, and the plug-and-play method in the following sub-sections. Details about how the generation experiments are setup and evaluated are given at the end of the section.

4.2.1 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 4.1). 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 $\mathbf{y}$ 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:

$$\texttt{Attention}(Q, K, V) = \texttt{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \tag{4.1}$$

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 4.1 for an schematic illustration of the Transformer's structure described above.

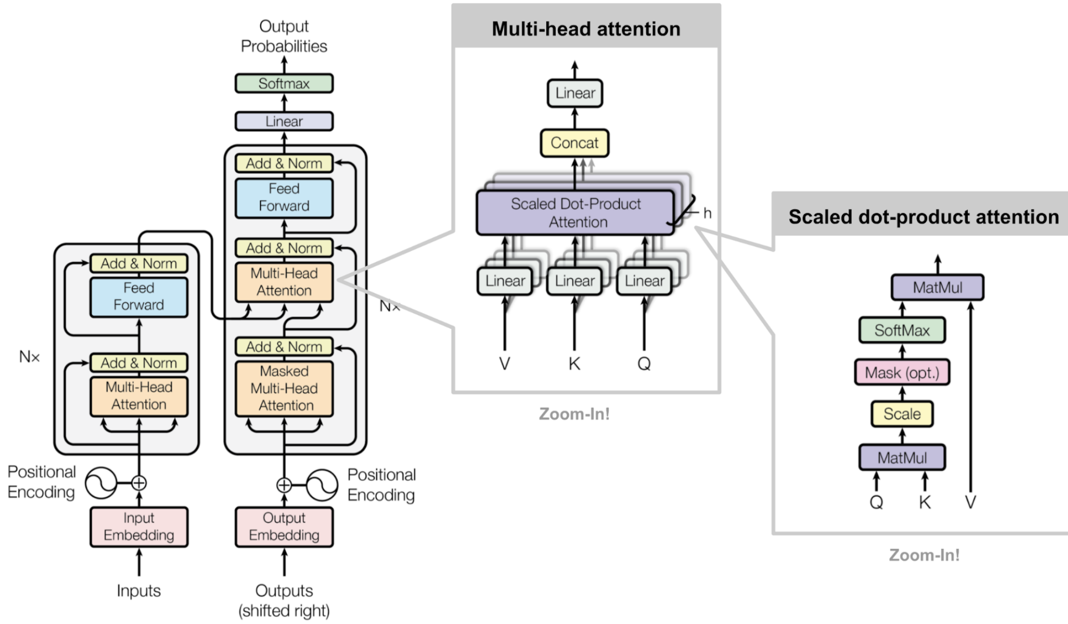


Figure 4.1: An overview of the full Transformer model architecture. *Collated image source:* Fig. 17 in this blog post `https://lilianweng.github.io/lil-log/2018/06/24/attention-attention.html`. *Original image source:* Figures 1 and 2 in Vaswani et al. [2017]

4.2.2 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:

$$p(\mathbf{x}) = \prod_{i=1}^{n} p(x_i|x_0, ..., x_{i-1}). \tag{4.2}$$

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:

$$o_{t+1}, H_{t+1} = \texttt{LM}\left(x_t, H_t\right), \tag{4.3}$$

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.

4.2.3 Conversational response generation

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. More formally, 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

$$p(T|S) = \prod_{n=m+1}^{N} p(x_n|x_1, ..., x_{n-1}). \tag{4.4}$$

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.

4.2.4 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:

$$p(X|a) \overbrace{=}^{\text{Bayes' theorem}} \frac{p(X)p(a|X)}{p(a)} \propto p(X)p(a|X). \tag{4.5}$$

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:

$$\Delta H_t \leftarrow \Delta H_t + \alpha \frac{\nabla_{\Delta H_t} \log p(a|H_t + \Delta H_t)}{\|\nabla_{\Delta H_t} \log p(a|H_t + \Delta H_t)\|^{\gamma}} \tag{4.6}$$

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 4.6 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., $x_{t+1} \sim \tilde{p}_{t+1} = \texttt{softmax}\,(W\tilde{o}_{t+1})$.

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:

$$x_{t+1} \sim \frac{1}{\beta}\left(\tilde{p}_{t+1}^{\gamma_{gm}} p_{t+1}^{1-\gamma_{gm}}\right) \tag{4.7}$$

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 4.7 converges towards the unconditional language model as $\gamma_{gm} \to 0$, repetitive text generation can be avoided by decreasing the scaling term.

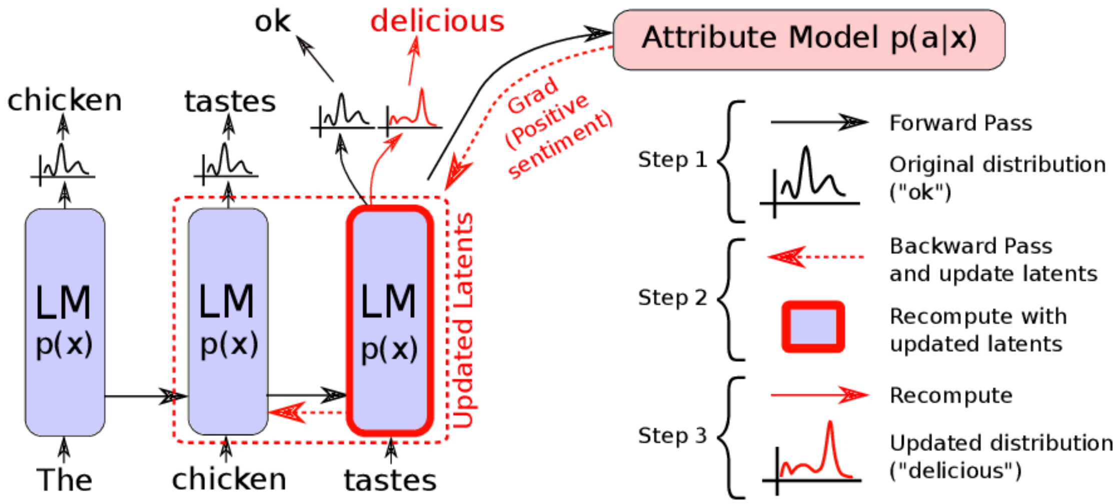


Figure 4.2: A schematic overview of the plug-and-play interaction between attribute model $p(a|\mathbf{x})$ and language model $p(\mathbf{x})$. *Original image source:* Figure 1 of Dathathri et al. [2020]

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 4.6 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. However, a fully interactive plug-and-play conversational model is out of the scope of this thesis.

4.2.5 Experimental details and evaluation

The workflow of my controlled text generation experiments can be divided into three phases, attribute model development, generation, and evaluation. The following paragraphs describe and motivate the steps and choices per phase. All experiments are conducted on an NVIDIA TitanRTX GPU.

Attribute model development During attribute model development, either a discriminator is trained on the dialogue data, or a bag-of-words is statistically constructed from the same corpus.

When training the discriminators, the dialogue dataset is randomly split into a training (90%) and test (10%) set. The frozen embeddings of either GPT-2-medium [Radford et al., 2019] or DialoGPT-medium [Zhang et al., 2020] are fed into trainable linear classifiers, seeking to distinguish between transcribed dialogue utterances from young (ages 19 to 29) and old (ages 50 and over) speakers. The discriminators are trained using Adam [Kingma and Ba, 2015] with a learning rate of $1 \cdot 10^{-4}$ and default values for all other parameters, with a maximum sequence length of 512 tokens, for 20 epochs, and a batch size of 64. The discriminator parameters are used of the epoch with the highest test accuracy.

A simple bag-of-words can also serve as an attribute model. Lists of unigrams are used as BoWs, because the PPLM setup is not compatible with lists of n-grams for $n > 1$, as it relies on perturbations at the unigram-level. Making a PPLM-system compatible with, e.g., trigrams, would amount of retraining the entire underlying language model (like GPT-2), thereby defeating the purpose of PPLM, i.e., leveraging large LMs for controllability, without incurring significant re-training costs. However, to continue the narrative between Experiment 1 and 2 (allowing the best-performing classifiers to inform decisions about generation), we use the best unigram's list of most informative features. This is a viable choice, because the unigram and trigram classifiers are on par (See Table 3.3).

I extend the methodology of Dathathri et al. [2020] that relies on curated wordlists, by applying two empirical approaches to extract wordlists from the dialogue corpus. An empirical approach has the benefit of being more reproducible, and not requiring a domain expert to manually curate a list. In the first approach, the BoW consists of the 100 most informative unigrams of the unigram classifier used during the text classification experiments (See Table 3.3 for the results). The most informative unigrams per age groups are deemed the most distinguishing features by the unigram classifier. They could therefore be used to make sensible perturbations to a language model's output, yielding more effective control.

The second method of wordlist extraction is fully frequency-based, these setups are labeled FB in Table 4.1. The goal of this extraction method is to yield two distinct sets of words that are representative of each age group's language. The frequency-based wordlists per age group are constructed from the *imbalanced* dialogue dataset as follows: Per age group, all unique words are ordered by frequency of occurrence in the corpus. For both ordered lists of word counts, the most frequent words are kept that account for at least 85% of the cumulative probability mass of the full age-specific distribution of words. Then, the words are removed that appear in both lists (i.e., the overlapping set of words is discarded). Of the resulting two non-overlapping ordered lists of words and their numbers of occurrences, only the words are kept that account for at least 85% of the respective wordlist's summed occurrences. The resulting lists consist of 56 (young), and 92 (old) words. The 85-th percentile cutoff points are chosen to yield wordlists of similar lengths as those used by Dathathri et al. [2020]. Both pairs of wordlists are included Section A.1.

Generation Controlled text generation experiments are performed using PPLM-setups that differ with respect to **(1)** pre-trained language model (GPT-2 or DialoGPT), **(2)** type of attribute model (BoW or discriminator), **(3)** attribute (young, old, or uncontrolled), and **(4)** whether the model was prompted or not. An unprompted model conditions its generated output on the `<|endoftext|>` token. BoW-based configurations can also differ in their wordlist extraction method (most informative unigrams or frequency-based).

Every PPLM-setup generates 30 sequences per output length 6, 12, 24, 30, 36, 42, 48, 54, and 60 tokens. Sub-sample sizes of 30 are chosen to satisfy the Central Limit Theorem (CLT), making it possible to assume the sub-samples approximate normal distributions [CLT, 2008]. The results in Table 4.1 are averaged over $N = 30 \cdot 9 = 270$ samples. Note that perturbations of the base language model's output can affect the controlled sequence length, so the final sequence length may differ by a few tokens from the given one. All other PPLM-parameters are kept at their default values, as recommended by Dathathri et al. [2020].

Each reported PPLM-configuration represents the best initialization, if the term applies. The pre-trained language models are kept equal across configurations, as using different initializations of these large models is infeasible, and would defeat the purpose of PPLM. A BoW-based setup uses a single list of words as an attribute model, thereby not having random parameters to initialize. And finally, discriminator-based setups use comparatively small linear classifiers (a few hundred parameters), the initialization effects of which have been found to be negligible on performance.

[L: TODO - Motivate the choice of prompts]

Evaluation The generated sequences are evaluated along two main axes: fluency and control. Fluency refers to the degree to which a text passage appears natural, grammatical, and non-repetitive. Control is the extent to which the produced language resembles that of the attribute being controlled for. Evaluation is done using both automated metrics and human opinions. Fluency is measured automatically by perplexity (denoted ppl) with respect to a different language model, GPT-1 [Radford et al., 2018], expressed as

$$\text{ppl}(\mathbf{x}) = \exp\left\{-\frac{1}{t}\sum_{i}^{t}\ln p_\theta(x_i|x_{<i})\right\}. \tag{4.8}$$

x represents a sequence of tokens, t is sequence length, x_i is the i-th token, and θ denotes GPT's parameters. Perplexity is a measure of a language model's uncertainty when posed with the task of predicting a succession of words. Assuming a language model to be a reliable representation of relationships within a natural language, low perplexity can serve as a rough proxy for fluency of a text. However, a major caveat of perplexity is that it only measures uncertainty w.r.t. one language model, making it less generalizable. To slightly reduce this effect, we choose to evaluate perplexity with respect to a different language model than the one used for generation (GPT-2 or DialoGPT).

Furthermore, the normalized number of distinct unigrams (Dist-1), bigrams (Dist-2), and trigrams (Dist-3), are used as measures of text diversity. Experiment 1's best BERT-classifier's classification accuracy on a set of sequences generated by a single generation model is used as an automated measure of attribute control. It can be seen as a proxy for control, because it indicates how resemblant of an age group's vernacular a generated text is deemed to be.

Two types of baselines are used when evaluating text generation performance: a pre-trained model baseline, and a corpus-specific baseline. The pre-trained model baseline refers to the uncontrolled language model setting, being either GPT-2 or DialoGPT. Therefore all controlled generation models that use GPT-2 as their language model, should be compared to the uncontrolled GPT-2 baseline. The same holds for DialoGPT. The second type of baseline combines the underlying language model with a bag-of-words consisting of the 100 most common words in the balanced dialogue corpus, irrespective of age. This setting is included to give an indication of how biased the balanced BNC's frequently occurring words might be towards a specific age group.

[L: TODO - Describe human evaluation]

4.3 Controlled text generation performance

Table 4.1 reports the automated evaluation results of our controllable text generation models. It can be seen that uncontrolled GPT-2 baseline has a slight bias towards generating "young-sounding" language (57.5% accuracy). Furthermore, it appears that perturbing GPT-2's output distribution with the 100 most common words across all ages results in a slight de-biasing of the generated text (54.1% accuracy). Achieving detectable control seems possible, because all GPT-2-based models surpass both baselines in terms of accuracy, with the exception of both BoW-setups using the 100 most informative unigrams.

Frequency-based BoW-models outperform those using the most informative unigrams, as illustrated by their higher average accuracy (66.75% versus. 53.1%), and lower average perplexity (27.48 versus 27.90). Discriminator-based models achieve noticeably better accuracies, with an average improvement of 8.45% over the best performing BoW-based models. However, discriminator-based models do show more signs of disfluency and repetitiveness compared to the BoW-models, as depicted by the worse perplexities and Dist-$n|_{n=1,2,3}$ scores.

The accuracies of our uncontrolled DialoGPT baseline (78.1%) and the 100MCW baseline (80.7%), suggest that DialoGPT is heavily biased towards producing young-sounding language. This can be attributable to DialoGPT having been fine-tuned on Reddit threads, as the majority of Reddit users are between the ages 20 and 29 [1] [Zhang et al., 2020]. DialoGPT's strong propensity for generating younger sounding language makes it a less desirable choice for our human evaluation experiments, because it requires non-standard parameter settings to produce detectably older sounding text.

Overall, the results show that, for most models, a plug-and-play approach to controlling generated dialogue responses to possess detectable age-specific linguistic features is achievable. The most promising models being either discriminator-based, or frequency-based bag-of-words models. Discriminator-based models achieve more detectable levels of control than their BoW-based counterparts, at the cost of perplexity and repetitiveness. This could be attributable to the more complex activation-space updates that are used by discriminator-models. Furthermore, GPT-2's preference to generate young-sounding language is severely less pronounced than that of DialoGPT, making it easier to control, given equal parameter settings.

[1] `https://www.statista.com/statistics/1125159/reddit-us-app-users-age/`

Model	**ppl.** ↓ better	**Dist-1** ↑ better	**Dist-2** ↑ better	**Dist-3** ↑ better	**Acc.** ↑ better
GPT-2 baseline	29.60 (±17.57)	**0.89** (±0.09)	**0.93** (±0.05)	0.88 (±0.10)	57.5%
GPT-2 100MCW baseline	27.80 (±16.44)	0.83 (±0.12)	0.91 (±0.06)	0.88 (±0.09)	54.1%
$\text{B}_{Y,FB}$	28.16 (±14.52)	0.87 (±0.10)	0.92 (±0.06)	0.88 (±0.10)	69.3%
$\text{B}_{O,FB}$	26.79 (±8.89)	0.88 (±0.09)	0.92 (±0.05)	0.88 (±0.10)	64.2%
$\text{B}_{Y,100MIU}$	29.16 (±14.91)	**0.89** (±0.09)	0.92 (±0.06)	0.88 (±0.11)	52.5%
$\text{B}_{O,100MIU}$	26.63 (±8.36)	0.87 (±0.10)	0.92 (±0.07)	0.88 (±0.10)	53.7%
$\text{D}_{Y,GPT2}$	31.95 (±14.29)	0.82 (±0.17)	0.87 (±0.14)	0.83 (±0.16)	**77.7%**
$\text{D}_{O,GPT2}$	33.63 (±24.40)	0.80 (±0.18)	0.87 (±0.11)	0.81 (±0.21)	72.7%
DGPT baseline	35.20 (±10.01)	0.87 (±0.11)	0.90 (±0.07)	0.87 (±0.08)	78.1%
DGPT-100MCW	35.64 (±9.72)	0.86 (±0.10)	0.90 (±0.06)	0.87 (±0.08)	80.7%
$\text{D*}_{Y,DGPT}$	41.54 (±10.87)	0.91 (±0.11)	0.91 (±0.06)	0.86 (±0.09)	**84.1%**
$\text{D*}_{O,DGPT}$	38.16 (±10.77)	0.87 (±0.11)	0.91 (±0.06)	0.87 (±0.08)	55.6%

Table 4.1: Results of age-controlled language generation. Perplexity is perplexity w.r.t. GPT-1. Dist-n is number of distinct n-grams normalized by text length, as a measure of diversity. Acc. is the best BERT model's accuracy when classifying the row's samples.

- [L: Include examples of same original sequence being perturbed differently at the unigram-level between corresponding young and old BoW-based CTG. E.g., *I think you're a nice person (old)* vs. *I think you're a nice guy (young)*]

4.4 Controlled text generation analyses

By means of quantitative and qualitative analyses, we seek to study which relationships drive the grammatical quality and attribute relevance of generated output. We study the relationship between fluency and control, the effects of sequence length, and visualize attention mechanisms. Finally, we provide qualitative inspections of various cases, i.e., high versus low perplexity or assigned $BERT_{FT}$ probability, and a human evaluation of linguistic quality.

4.4.1 Quantitative analyses

[L: TODO - Add examples of generated sequences along with their model's configurations, age-group, etc. Similar to dialogue snippets earlier.]

Quantitative 0: the relationship between fluency and control

Figure 4.3 attempts to depict the relationship between fluency and control, as measured by perplexity and BERT's classification accuracy, respectively. On the y-axis, "mean accuracy" refers to the average fraction of generated sequences, controlled for young or old language, that are correctly labeled as such by Experiment 1's best BERT classifier. The bars around the averages in Figure 4.3 are 90% confidence intervals. It appears that increasing perplexity is slightly negatively correlated with accuracy. It is also clear from Figure 4.3 that uncertainty about prediction strongly increases for greater perplexity. These two observations indicate that sentences deemed less coherent by GPT-1 tend to be harder to classify by BERT$_{FT}$ with certainty. BERT$_{FT}$ is pre-trained and fine-tuned to pick up syntactic features from dialogue that can indicate a speaker's age. It is therefore plausible that structural deviations from proper syntax (i.e., high perplexity) can obfuscate the age-related linguistic signal BERT$_{FT}$ leverages. Finally, it seems that, on average, the discriminator-based models are more capable of producing correctly classifiable high-perplexity sentences. [L: Why would that be?] However, none of the differences between the average accuracies are statistically significant at the 10% level, so this conclusion should be taken tentatively.

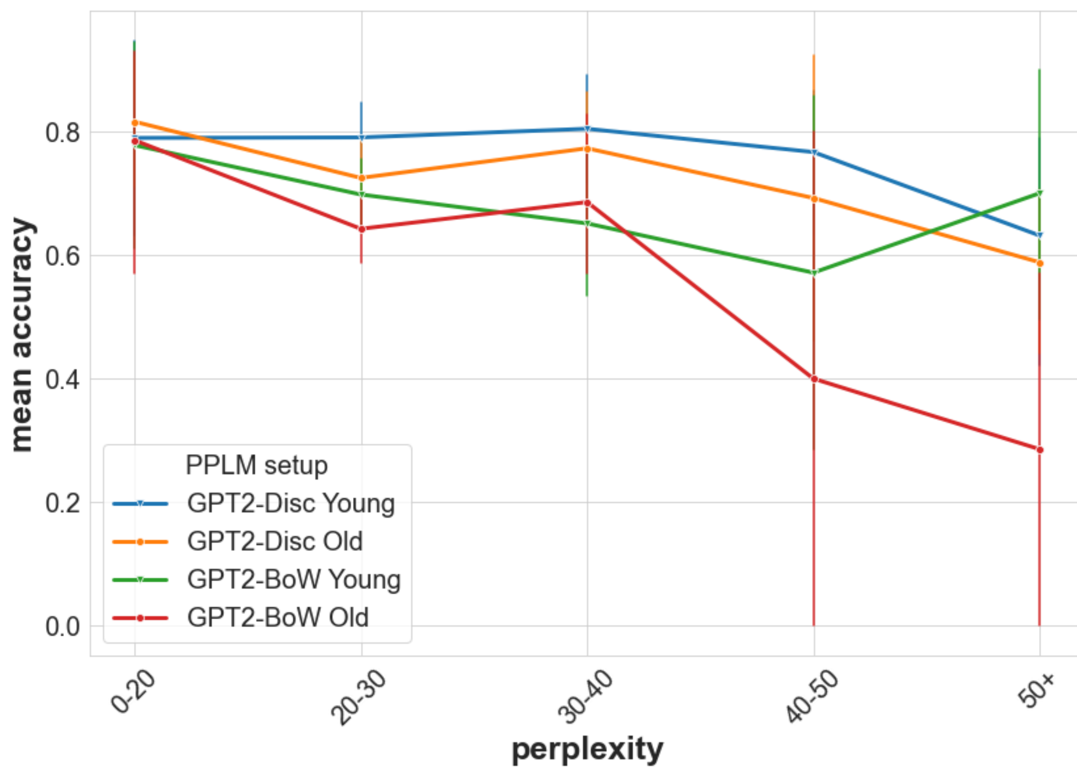


Figure 4.3: Fluency (perplexity, x-axis) versus control (BERT accuracy, y-axis). Bins chosen to be roughly same order of magnitude. The bars represent 90% confidence intervals. This plot is best viewed in color.

Quantitative 1: the effects of generated sequence length

- *Main question: how is generated sequence length related to fluency and control?*

- *Study the relationship between generated sequence length (measured in number of tokens) and automated evaluation metrics (i.e., perplexity, dist-n, and accuracy).*
- *For every metric and for (all?) models, plot sequence length on the x-axis, and the average metric with confidence intervals on the y-axis.*
- *Which patterns do you observe?*

Patterns:

- [L: NB: These are all patterns observed in the unprompted results. Once prompted results are in, we can interpret them as if length of dialogue response.]
- Increasing sequence length is correlated with decreasing perplexity. Longer sentences are deemed more coherent by GPT-2
- Increasing length seems very slighty positively correlated with average accuracy (and more uncertainty). I.e., longer sequences are, on average, easier to classify, though less precision.
- BoW-based models significantly more distinct w.r.t. unigrams than dirscrim-based.
- Overall, repetitiveness seems to increase as generated sequences become longer.
- The differences between young and old perplexity are smaller for BoW-based models than for Discrim-based. Same pattern holds for repetitiveness.

[L: NB: These are all patterns observed in the unprompted results. Once prompted results are in, we can interpret them as if length of dialogue response.]

The number of tokens being generated in an utterance seems to coincide with noticeable differences in our automated evaluation metrics. It is therefore important to get a clearer picture of how the various measures for fluency and control change for varying sequence lengths. Properly understanding this relationship can inform developers of adaptive dialogue systems about preserving output quality for responses of arbitrary lengths.

Figure 4.4 presents plots of the relationships between generated sequence length (on the x-axes) and average perplexity (Figure 4.4a), average BERT$_{FT}$ accuracy (Figure 4.4b), and average normalized number of distinct unigrams (Figure 4.4c), bigrams (Figure 4.4d), and trigrams (Figure 4.4e).

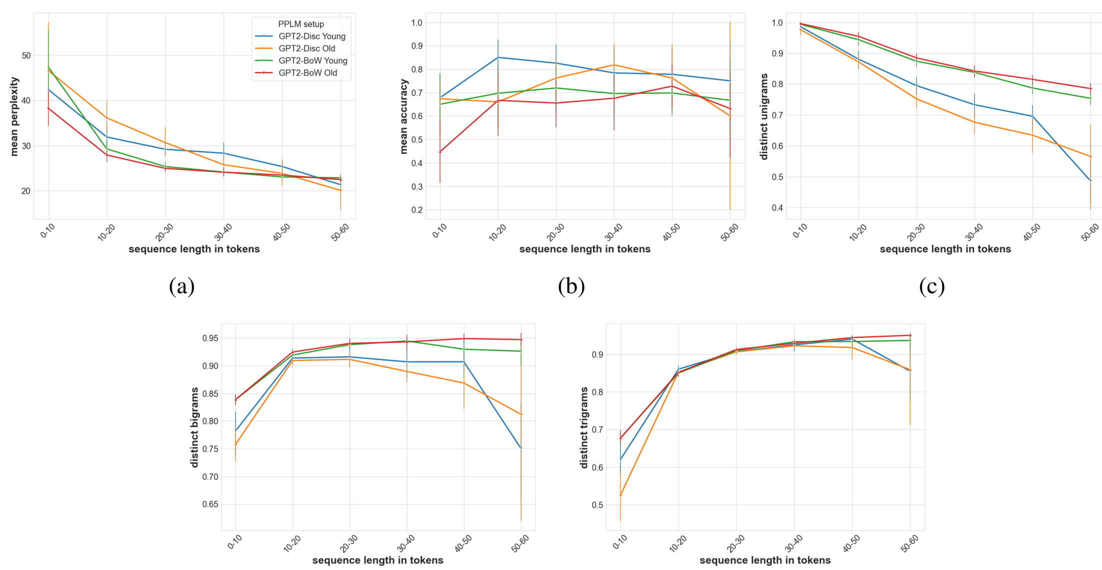
Starting with Figure 4.4a, it appears that, for all models, increases in generated utterance length coincide with decreases in perplexity. This is most likely attributable to the nature of calculating perplexity than generation properties of the models. Namely, perplexity essentially averages the sum of the negative exponentiated probabilities $p(\texttt{word}|\texttt{context})$, for every word in a sentence. Because the context increases with every successive word, and larger contexts typically result in less uncertainty, shorter sequences are often given unfairly high perplexities.

In Figure 4.4b, we can see that increasing length seems very slightly positively correlated with average accuracy (and more uncertainty). That is, longer sequences are, on average, slightly easier to classify, though with less precision. This is probably due to the fact that longer sentences contains more information to base predictions on.

Focusing on repetitive use of unigrams, it appears that BoW-based models generate significantly more diverse utterances than discriminator-based models.

Repetitiveness w.r.t. bigrams and trigrams seems to increase as generated sequences become longer.

Overall, the differences between young and old perplexity are smaller for BoW-based models than for Discriminator-based. Same pattern holds for repetitiveness.



(a) (b) (c)

(d) (e)

Figure 4.4: Various averaged automated evaluation metrics of generated sentences, plotted against increasing sequence length (x-axis), with 90% confidence intervals. The plots are best viewed in color.

Quantitative 2: the effects of PPLM-parameters on fluency and control

- *Plot and examine the relationship between fluency and control, and various PPLM-parameters (step-size, number of iterations, temperature, top k, gamma, KL-scale).*

- *Which patterns do you observe?*
- [L: How much sense does it make to study this, though? Is that the purpose of my thesis? Hasn't this been studied enough in the PPLM-paper? Which parameters do I choose?]

Quantitative 3: BERT-classifier visualizations. Or (Dialo)GPT visualizations.

- ***NB:** This is more relevant to the classification experiments, than to the controlled generation experiments.*
- *Use BertViz to visualize what parts of sequences BERT's transformer heads and neurons are focusing on.*
- `https://github.com/jessevig/bertviz`
- `https://towardsdatascience.com/openai-gpt-2-understanding-language-generation-throug`

4.4.2 Qualitative analyses

Qualitative 1: summary statistics and qualitative inspection of various cases

- *Similar to (error)case analyses of Experiment 1.*
- *Provide summary statistics and (**qualitative**) inspection of generated sequences per case.*
- *Cases could be: (1) sequences with low, average, high, or very-high perplexity. (2) (in)correctly classified generated sequences.*
- *What patterns do you observe among, e.g., misclassified sequences with low perplexity?*
- *Provide table of examples per case and age-group. Similar to table 3.4*

Qualitative 2: Human evaluation of fluency, grammaticality, and relevancy

- *Generate and sample text passages for a variety of model-configurations and age-groups.*
- *Have a group of human participants rate these sequences on a scale from 1 to 5 for their (1) fluency, (2) grammaticality, (3) relevance to the prompt (if there is one)*
- *Average the ratings, and compare the human evaluation metrics to the automated evaluation metrics reported in Table 4.1*

Chapter 5

Discussion

Discussion points about classification

- Can any ML architecture pick up signals from 1-6 token sequences? (See workshop paper submission feedback).
- age-related linguistic features that inform classification lie more at the syntactic level than at the lexical level.
- [L: Re-read the relevant sections of the workshop paper submission]

Discussion points about generation

- What are the effects of prompts on generation? [L: This should probably be an analysis question.]
- What are the limitations of your setup?
- Does this make the world better? How can this help people? –> It can help personalize virtual assistants (especially useful for new speakers of a language. E.g., the difference between young/informal/spoken French and French that is taught in school and courses is large. User-age personalization can adapt use of language of virtual assistants to variant of language spoken by user.)
- What are the dangers of these methods? –> Read up on paper by Ebru et al
- What are interesting future research directions?
- My research is a promising step towards the development of personalized virtual assistants.

- Future research idea: adapting the PPLM-setup to work with n-gram lists for arbitrary n.
  - Finding a way to by-pass the need to retrain GPT-2 for arbitrary n-grams
- Future research idea: real-time interactivity of age-adaptive conversational systems. I.e., a pipeline that (1) "starts off neutral", (2) classifies user's age based on minimal amount of utterances, (2*) uses bayesian modelling or reinforcement learning to constantly update belief, (3) adapts use of language to perceived user age.

Chapter 6

Conclusion

- Sum up what this thesis/research entailed.
- Repeat most salient conclusions and insights of thesis.
- ...

Bibliography

Central Limit Theorem, pages 66–68. Springer New York, New York, NY, 2008. ISBN 978-0-387-32833-1. doi: 10.1007/978-0-387-32833-1_50. URL https://doi.org/10.1007/978-0-387-32833-1_50.

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.

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. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

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.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

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 J. Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015. URL `http://arxiv.org/abs/1412.6980`.

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`.

M. Li, K. J. Han, and S. Narayanan. Automatic speaker age and gender recognition using acoustic and prosodic level information fusion. *Computer Speech & Language*, 27(1):151–167, 2013.

D. C. Liu and J. Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.

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.

A. Nguyen, J. Clune, Y. Bengio, A. Dosovitskiy, and J. Yosinski. Plug & play generative networks: Conditional iterative generation of images in latent space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2017.

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.

M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

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.

M. Wolters, R. Vipperla, and S. Renals. Age recognition for spoken dialogue systems: Do we need it? In *Tenth Annual Conference of the International Speech Communication Association (Interspeech)*, 2009.

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.

Z. Zhang, S. Ren, S. Liu, J. Wang, P. Chen, M. Li, M. Zhou, and E. Chen. Style transfer as unsupervised machine translation. *CoRR*, abs/1808.07894, 2018. URL `http://arxiv.org/abs/1808.07894`.

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

Appendix A

Supplementary material

A.1 Wordlists for BoW-based approaches

[L: TODO - Censor foul language?]

100 most informative unigrams - young (19-29) um, cool, shit, hmm, uni, cute, tut, massive, awesome, gym, bitch, lol, grand, pizza, like, excited, yawn, Korea, cigarette, fuck, fairness, Jesus, annoying, Facebook, quicker, definitely, guess, Sunderland, oo, wanna, mountain, scared, piss, love, miss, Middlesbrough, mhm, specifically, ooh, website, roundabout, photo, nope, blanket, management, ridiculous, mental, pregnant, beers, hate, log, fucking, cry, cheaper, skinny, plural, burger, hilarious, hint, drunk, fridge, cousin, coke, genuinely, James, mates, smaller, option, balance, saving, basically, leather, nev, shut, frig, mate, yay, invite, maid, nickname, badly, garlic, CD, jokes, Uzbekistan, boyfriend, date, added, Manchester, blah, shitty, lang, tempted, stadium, wee, eh, baking, city, honestly, exam

100 most informative unigrams - old (50 plus) ordinary, Chinese, wonderful, yes, tend, father, photographs, vegetables, hospice, operation, shed, pension, areas, mother, hanging, hospices, glasses, chap, anyhow, tank, surgery, container, cheers, born, church, pain, several, workshop, right, horses, building, extraordinary, vegetarian, biscuit, americano, engine, luck, paint, emperor, lipsy, trombone, occasional, supper, lord, architect, council, roast, schools, bath, asbestos, endometrial, concrete, poodle, recall, diabetes, misty, report, heavens, enormous, lawn, potatoes, email, junk, scabies, mousse, Ebola, churches, sewing, plants, rackets, marmalade, engineering, furniture, photograph, sandwiches, unemployment, xylophone, Piccadilly, flu, claim, arab, nineteen, forgotten, sensible, blancmange, spencer, yards, emails, yellow, scruffy, fungi, garden, boiler, lodge, mostly, Robson, tricky, shark, robin, contracture

Frequency-based young (19-29) um, shit, cool, fucking, definitely, guess, friends, everyone, literally, dad, sounds, weekend, loads, watch, fair, fuck, amazing, friend, ha, huh, hate, fun, stay, girl, holiday, blah, hours, uni, month, horrible, massive, Friday, stupid, film, parents, thirty, spend, mate, honest, change, hope, yourself, annoying, wear, wait, ridiculous, anyone, Saturday, tea, dinner, sit, crazy, hell, pound, nine, expensive

Frequency-based old (50 plus) building, may, water, mother, perhaps, door, lots, business, cancer, area, although, worked, open, cut, number, under, young, nineteen, everybody, garden, church, case, shop, children, certainly, set, coffee, email, gave, white, along, doctor, hear, often, possibly, group, father, outside, wonderful, taken, seem, places, green, given, hand, early, women, space, front, language, dear, light, huge, supposed, country, hospital, otherwise, asked, putting, bits, gosh, wall, woman, almost, particularly, across, word, age, rest, flat, turned, decided, finished, needed, red, bin, hospice, running, slightly, its, middle, local, percent, Chinese, paper, check, high, milk, piece, near, nobody, usually

A.2 Where to put these?

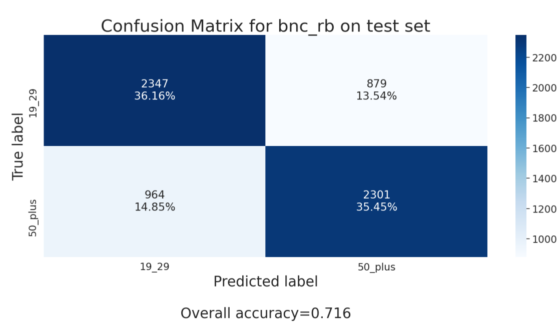


Figure A.1: Confusion matrix BERT age classifier on balanced BNC **test** set.

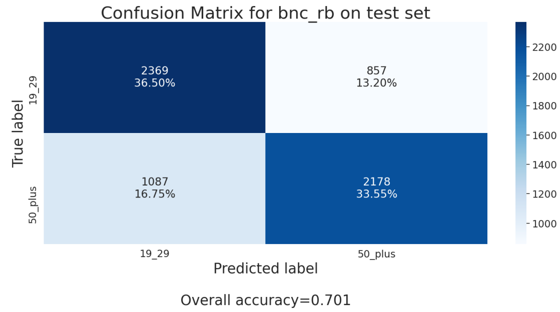


Figure A.2: Confusion matrix LSTM age classifier on balanced BNC **test** set.

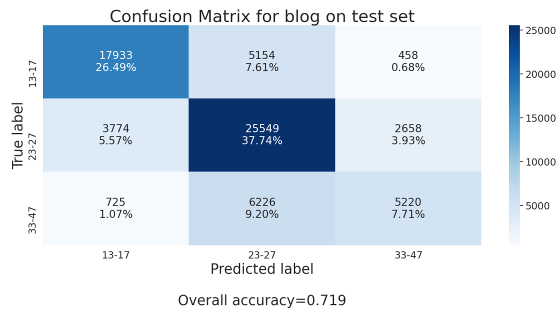


Figure A.3: Confusion matrix bi-LSTM age classifier on blog corpus **test** set.

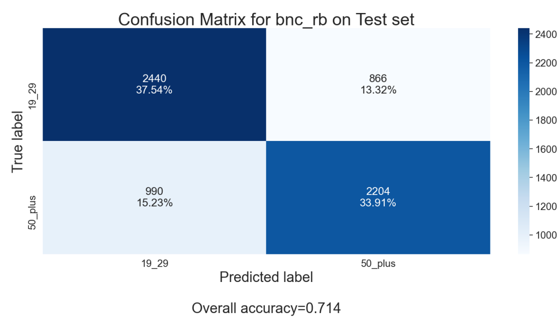


Figure A.4: Confusion matrix for best trigram age classifier on **balanced** BNC **test** set.

A.3 Age discrimination on the imbalanced British National Corpus

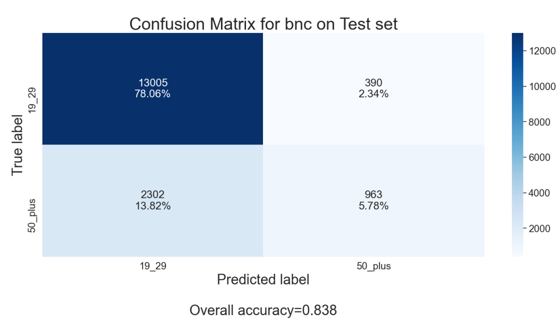


Figure A.5: Confusion matrix for best bigram age classifier on BNC test set.

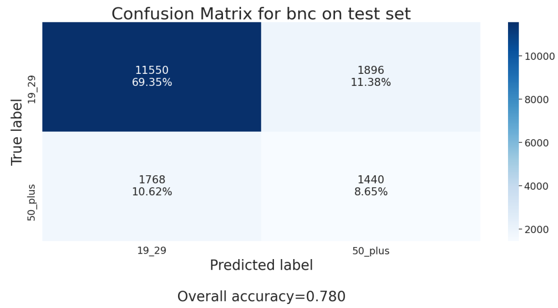


Figure A.6: Confusion matrix bi-LSTM age classifier on BNC test set.

19-29		50+	
coef.	**n-gram**	**coef.**	**n-gram**
-3.19	um	2.29	yes
-2.91	cool	2.21	wonderful
-2.70	s**t	1.91	building
-2.25	cute	1.86	right right
-2.15	uni	1.80	something like
-2.14	hmm	1.73	garden
-1.97	wanna	1.69	right
-1.93	f**k	1.68	ordinary
-1.91	like	1.67	shed
-1.85	massive	1.63	operation
-1.83	yeah course	1.58	born
-1.81	love	1.57	mother
-1.79	tut	1.55	photographs
-1.74	b***h	1.51	email
-1.68	like oh	1.08	anything like

Table A.1: [L: Excluding stopwords.] For each age group, top 15 most informative n-grams used by the trigram model. **coef.** is the coefficient (and sign) of the corresponding n-gram for the logistic regression model: the higher its absolute value, the higher the utterance's odds to belong to one age group. * indicates masking of foul language.

Model	**Accuracy** ↑ better	$\boldsymbol{F}_1^{(19-29)}$ ↑ better	$\boldsymbol{F}_1^{(50+)}$ ↑ better
Random	0.500	0.500	0.500
unigram	0.702 (0.006)	0.713 (0.006)	0.690 (0.006)
bigram	0.703 (0.006)	0.713 (0.005)	0.693 (0.008)
trigram	0.709 (0.007)	**0.718** (0.007)	0.700 (0.008)
LSTM	0.696 (0.005)	0.689 (0.018)	0.701 (0.016)
BiLSTM	0.684 (0.007)	0.688 (0.018)	0.679 (0.016)
BERT_{frozen}	0.673 (0.005)	0.679 (0.013)	0.667 (0.018)
BERT_{FT}	**0.710** (0.006)	0.717 (0.007)	**0.703** (0.014)

Table A.2: Dialogue dataset [L: Excluding stopwords]. Test set results averaged over 5 random initializations. Format: *average metric (standard error)*. Values in **bold** are the highest in the column; in blue, the second highest.

Model	**ppl.** ↓ better	**Dist-1** ↑ better	**Dist-2** ↑ better	**Dist-3** ↑ better	**Acc.** ↑ better
Baseline***	27.45 (±7.27)	0.90 (±0.10)	0.92 (±0.05)	0.86 (±0.09)	57.5%
B_{100MCW}***	26.68 (±8.77)	0.89 (±0.10)	0.92 (±0.05)	0.86 (±0.09)	51.7%
$\text{B}_{Y,FB}$	27.11 (±7.45)	**0.91** (±0.09)	**0.92** (±0.04)	**0.87** (±0.09)	68.3%
$\text{B}_{O,FB}$	25.99 (±6.41)	0.88 (±0.11)	0.92 (±0.05)	0.86 (±0.09)	62.5%
$\text{B}_{Y,100MIU}$	28.48 (±11.96)	0.88 (±0.12)	0.91 (±0.06)	0.86 (±0.10)	69.2%
$\text{B}_{O,100MIU}$	**25.57** (±7.44)	0.88 (±0.11)	0.92 (±0.05)	**0.87** (±0.09)	58.3%
$\text{D}_{Y,GPT2}$	33.02 (±12.24)	0.85 (±0.16)	0.89 (±0.07)	0.83 (±0.12)	73.9%
$\text{D}_{O,GPT2}$	32.86 (±18.08)	0.80 (±0.21)	0.84 (±0.13)	0.79 (±0.19)	63.3%

Table A.3: [L: Excluding stopwords.] Results of age-controlled language generation. Perplexity is perplexity w.r.t. GPT-1. Dist-n is number of distinct n-grams normalized by text length, as a measure of diversity. Young and old accuracy are the assigned probabilities of belonging to the young or old age categories.