Universidade Estadual de Campinas
Instituto de Computação

Jader Martins Camboim de Sá

Curricular Transfer Learning for Complex Grammar Acquisition in Language Models

Transferência de Aprendizado Curricular para Aquisição de Gramáticas Complexas em Modelos de Linguagem

CAMPINAS
2022

Jader Martins Camboim de Sá

Curricular Transfer Learning for Complex Grammar Acquisition in Language Models

Transferência de Aprendizado Curricular para Aquisição de Gramáticas Complexas em Modelos de Linguagem

Dissertação apresentada ao Instituto de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Dissertation presented to the Institute of Computing of the University of Campinas in partial fulfillment of the requirements for the degree of Master in Computer Science.

Supervisor/Orientador: Prof. Dr. Leandro Aparecido Villas
Co-supervisor/Coorientador: Prof. Dr. Julio Cesar dos Reis

Este exemplar corresponde à versão final da Dissertação defendida por Jader Martins Camboim de Sá e orientada pelo Prof. Dr. Leandro Aparecido Villas.

CAMPINAS
2022

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Silvania Renata de Jesus Ribeiro - CRB 8/6592

Sa11c

Sá, Jader Martins Camboim de, 1994-
Curricular transfer learning for complex grammar acquisition in language models / Jader Martins Camboim de Sá. – Campinas, SP : [s.n.], 2022.

Orientador: Leandro Aparecido Villas.
Coorientador: Julio Cesar dos Reis.
Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Aprendizado profundo. 2. Processamento de linguagem natural (Computação). 3. Aquisição de linguagem - Processamento de dados. I. Villas, Leandro Aparecido, 1983-. II. Reis, Julio Cesar dos, 1987-. III. Universidade Estadual de Campinas. Instituto de Computação. IV. Título.

Informações Complementares

Título em outro idioma: Transferência de aprendizado curricular para aquisição de gramáticas complexas em modelos de linguagem
Palavras-chave em inglês:
Deep learning
Natural language processing (Computer science)
Acquisition of language - Data processing
Área de concentração: Ciência da Computação
Titulação: Mestre em Ciência da Computação
Banca examinadora:
Leandro Aparecido Villas [Orientador]
Thiago Henrique Silva
Roberto de Alencar Lotufo
Data de defesa: 20-12-2022
Programa de Pós-Graduação: Ciência da Computação

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0002-4457-9996
- Currículo Lattes do autor: http://lattes.cnpq.br/1926004700254677

Universidade Estadual de Campinas
Instituto de Computação

Jader Martins Camboim de Sá

Curricular Transfer Learning for Complex Grammar Acquisition in Language Models

Transferência de Aprendizado Curricular para Aquisição de Gramáticas Complexas em Modelos de Linguagem

Banca Examinadora:

- Prof. Dr. Leandro Aparecido Villas
  IC/UNICAMP
- Prof. Dr. Thiago Henrique Silva
  DAINF/UTFPR
- Prof. Dr. Roberto de Alencar Lotufo
  FEEC/UNICAMP

A ata da defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade.

Campinas, 22 de agosto de 2022

Acknowledgment

I thank my parents, Judá and Elizete, for all the emotional, operational, and financial support I always had to dedicate myself to my studies. My supervisors, Leandro and Julio, were always solicitous and helpful in producing the research and this dissertation. And to the other research colleagues, Rafael, Matheus, Paula, Henrique, and Andreis, who contributed directly or indirectly to the production of this work.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. Ci&T also financed this research, so I acknowledge them for their support.

Resumo

O uso de modelos de transferência de aprendizado em problemas de processamento de linguagem natural tornou-se uma prática padrão na academia e na indústria. Isso permite ajustar modelos pré-treinados em um conjunto de dados menor, obtendo maior generalização do que abordagens sem pré-treinamento. No entanto, quando a distribuição entre a tarefa pré-treinamento e a tarefa-alvo varia significativamente, por exemplo, ao considerar ambientes informais e gramática especializada, esses ganhos tendem a ser diminuídos. Nesse contexto, uma quantidade maior de dados na tarefa de destino torna-se necessária. A obtenção de dados supervisionados é complexa e cara. Para algumas aplicações, torna-se inviável a operacionalização. Por isso, várias pequenas empresas se limitam a usar essa tecnologia no dia a dia.

Esta dissertação de mestrado propõe uma nova abordagem para a aprendizagem por transferência curricular na qual tarefas intermediárias são criadas em uma sequência de etapas de treinamento. Nossa abordagem é guiada pelo "data hacking" que permite uma adaptação mais gradual entre o pré-treinamento e as distribuições alvo. Isso tem o potencial de reduzir significativamente os custos de geração de dados anotados. Nossa contribuição revela que a inserção de tarefas intermediárias no treinamento fim-a-fim de Sistemas de Diálogo Orientados a Tarefas permite uma melhor otimização do modelo final. Como consequência, uma avaliação de alto nível deste agente. Para avaliar essa metodologia, recorremos a dados bem estabelecidos na academia e projetamos um conjunto de dados para experimentação. Nesta metodologia, observamos uma melhora significativa em comparação com outras abordagens de pré-treinamento conhecidas.

Abstract

The use of transfer learning models in natural language processing problems has become standard practice in academia and industry. This allows adjusting pre-trained models in a smaller dataset, obtaining greater generalization than approaches without pre-training. However, when the distribution between the pre-training task and the target task varies significantly, for example, in considering informal environments and specialized grammar, these gains tend to be diminished. In this context, an enormous amount of data in the target task becomes necessary. Obtaining supervised data is complex and costly. For some applications, it becomes unfeasible to operationalize. Therefore, several small companies are limited to using this technology in their day-to-day operations.

This M.Sc. Thesis proposes a novel approach to curriculum transfer learning in which intermediate tasks are created in a sequence of training steps. Our approach is guided by "data hacking" that allows a more gradual adaptation between the pre-training and target distributions. This has the potential to significantly reduce the generation costs of annotated data. Our contribution reveals that the insertion of intermediate tasks in end-to-end Task-Oriented Dialog Systems training allows better optimization of the final model. As a consequence, a high-level evaluation of this agent. To evaluate this methodology, we resorted to well-established data in the academy and designed a dataset for experimentation. In this methodology, we observed a significant improvement compared to other known pre-training approaches.

List of Figures

List of Tables

List of Acronyms

NLP Natural Language Processing

CTL Curricular Transfer Learning

TODS Task-Oriented dialog Systems

LM Language Model

AI Artificial Intelligence

RL Reinforcement Learning

NER Named Entity Recognition

NN Neural Network

MDP Markov Decision Process

DST Dialog State Tracking

DP Dialog Policy

RG Response Generation

NLG Natural Language Generation

NLU Natural Language Understanding

Contents

Chapter 1

Introduction

1.1 Context and Motivation

The industry's digital revolution started decades ago, with the advent of the Internet and personal computers. This revolution seems far from an end due to its continuous insertion into the most diverse areas, like health, agriculture, and mobility (Bojanova, 2014). As a general trend, with the increased usage of edge devices, several users are abandoning traditional counter services and adopting pure virtual means to solve their problems, like websites or apps (Nedelkoska and Quintini, 2018; Helbing, 2019; Acemoglu and Restrepo, 2020).

These interface solutions with rich user experience could solve the needs of services and stand as the primary solution for many applications. However, they are not fully adequate for complex tasks, which generally require a time-consuming interaction with searching and clicking counter-intuitive interfaces in web pages (Adamopoulou and Moussiades, 2020). Examples of these complex interactions are solving a technical problem with a credit card or returning some defective product. This trend increased the demand for enhanced interfaces designed for improved Human-Computer interaction experiences.

Natural language communication can be considered the most acceptable, flexible, and richest interface when dealing with humans (Foster, 2019). Companies adopt chat interfaces where employees can directly communicate with customers via text messages to solve their needs. At the same time that this could be an alternative adopted in some business models, this kind of service has high costs (Cugini et al., 2007; Huang et al., 2020) and could not be affordable by many others. In recent years, automating this process using expert systems has become a hot topic for the industry and the academy (Huang et al., 2020). In this context, agents capable of rich natural language functionalities emerged as a research area to address this problem.

The field of Natural Language Processing (NLP) and Computational Linguistics enables computers to understand human language automatically (Russell and Norvig, 2010). Whereas early attempts focused purely on complex rule-based approaches, none of those models could work in unrestricted environments with high effectiveness. Statistical modeling has become the standard method for obtaining those rules that generalize for many contexts (Jurafsky and Martin, 2020). When equipped with several layers and adequate blocks for language representation, statistical models might acquire a human-like performance while being less challenging to engineer (Eisenstein, 2019). Furthermore, at the top of those models, Language Models (LMs) stands as the standard technique to deploy state-of-the-art intelligent systems capable of communicating.

Several fields in Artificial Intelligence (AI) still require supervised data to accomplish human-level behavior. Differently, language modeling enables models to attain high standards in NLP with unsupervised data, which has little to no cost for data acquisition; then apply those models with a high bias to other fields, with few or zero-shot learning (Brown et al., 2020). This kind of approach revolutionized the field of deep learning because it enabled searching for better learning architectures that are stable and scalable instead of producing better context-specific data sets.

TODS is a conversational agent that interacts through natural language with customers and can accomplish tasks by communicating with other computational systems. This agent is one of the best cost-effective solutions for assisting customers in complex scenarios. A customer can freely describe a problem (s)he wants to solve in natural textual language, and the agent can execute the desired activity with the help of artificial intelligence and engineering techniques. The agent should not only perform those resolutions but needs to behave like a human to keep customers engaged and satisfied (Roller et al., 2020).

1.2 Problem Characterization

State-of-the-art dialog systems are trained in unsupervised data by language modeling (Raffel et al., 2020), exploring the massive capabilities of scaled LMs. In recent news, one of those models, LaMDA (Thoppilan et al., 2022), trended as being conscious by deceiving humans on how well it could behave "humanly" (Luscombe, 2022). In general cases, it could solve problems in NLP out-of-the-box. It is not reasonable for complex and context-dependent problems, and the model should be fine-tuned for this task.

The TODS task, encoded as an end-to-end sequence for language modeling, presents an unusual structure that is not present in the data used for pre-training LMs. Figure 1.1 exemplifies this observation. The encoded sentence is a regular grammar for the meta-structure, where utterances have a conversation bias, with a distribution of words drifting from the general domain. We have a classification and Named Entity Recognition (NER) tasks for the belief state where the model infers the user's intent and extracts slot values. Finally, it learns an action to take, which shapes the following response.

Although the grammar for encoding the TODS seems feasible when each task is observed individually, the meta-generalization required for the learner to acquire this complex grammar simultaneously is not easily doable with restricted data (Su et al., 2021). Those over-parametrized models tend to overfit the supervised data, leading to poor generalization and prompting undesirable effectiveness in real-world applications. In particular, for the case of TODS, which acts in a sequential decision that could propagate errors over the process. This requires many more expensive annotated examples compared to standard NLP tasks.

While the unsupervised data generally have derisive costs, supervised data is costly

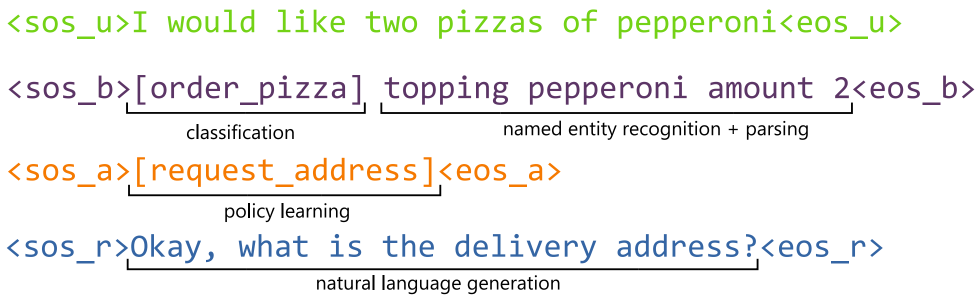


Figure 1.1: Example of NLP tasks presented in an encoded end-to-end TODS for a pizza ordering chatbot.

to build (Budzianowski et al., 2018; Byrne et al., 2019). For small business models with specific needs and context, conversational agents present as a non-affordable solution. This context points out that many businesses could not benefit from the scalability and optimized operationalization this agent could provide.

Framing a problem in easier abstractions is an old pedagogical approach that was also successful in training Deep Neural Networks (NNs) (Bengio et al., 2009). It is challenging the training conversational agents that should perform complex tasks while having enough annotated data for detecting statistical patterns. Our research investigates how to explore biases in varying tasks to improve and reduce the engineering costs of conversational agents.

1.3 Objectives and Developed Approach

Inspired by pedagogical and optimization literature, a curriculum approach presents examples to learners that gradually increase in complexity or difficulty. This allows the learner to focus on acquiring the basic skills in the initial steps and helps to learn later posed examples (Bengio et al., 2009). By the same principle, we could guide the learner through a learning process by presenting not only examples but complete new tasks with increasing complexity, similar to how humans acquire knowledge over tasks. For example, teaching a kid basic arithmetic helps them later to learn differential calculus.

The transfer learning approach trains a model with an auxiliary task to artificially yield more data for the training process. In the sequential case, a source task, where the source task is similar to the target task, presents a considerable amount of data. We can use it as a pre-training phase, where later, this model with a better initialization presents an adequate starting point for fine-tuning the target task. Although pre-training significantly helps in the learning process, when the selected hypothesis deviates between source and target, the final model cannot present desirable effectiveness, and we need more expensive data in the target domain.

In this MS.c Thesis, we propose creating a pre-training curriculum named Curricular Transfer Learning (CTL), in which we consider intermediate tasks instead of a single source task. In these intermediate tasks, we acquire training data and simulate one or more properties presented in the final task. Given that the bias is more suited for the final task, the model can demand fewer update steps for the final task by having another pre-training step. This helps it be less prone to overfit and acquire a better generalization.

To simulate our target data, we use the structure present in online forums, especially for this study, we selected the TripAdvisor forum. In this kind of social media, a user starts a thread on a focused topic, like games, movies, etc. The thread creator receives many replies to its message based on what he is requesting.

To create data that resembles our TODS data, we use the discussed topic to simulate the intent classification task. As conversational forums have a distribution of words that follow dialog acts, we use the thread creator's message as utterances and other replying messages as responses.

We conducted this research in the context of a project funded by Ci&T. The critical concern in the project was the investigation and the development of humanized chatbots, the commercial name for TODS, to solve the needs of small businesses in Brazil. While first-generation chatbots that only allow prompted menu navigation are well-established in the industry. Fourth-generation conversational agents have no tooling and engineering for the deployment. So, one of the crucial points in this research project is to deploy a fourth-generation chatbot solution from the ground up to deepen the researchers' technique and create the required tools.

This project contributed to us to evaluate the applicability of our methodology. To this end, we simulate a real-world scenario to deploy a conversational agent in the context of pizza ordering.

1.4 Synthesis of Results

In this MS.c. thesis by analyzing the grammatical structures and properties of a sequence-encoded TODS dataset, we discovered a strategy to generate data that resembles the same properties of the addressed problem. So we propose a strategy to generate intermediate training data, which works as a curriculum in the grammar acquisition level for LMs.

By exploring variations in pre-training strategies, we found that a curriculum of grammar-encoded tasks helps initialize downstream TODS tasks and significantly improves the overall effectiveness of LMs both in direct optimization and high-level evaluation. We found in our experiments that via meta-learning, it is, without teaching explicit tasks, but showing general language skills we desire our model to acquire, the LM presents a better initialization and convergence in the out-of-sample data in desired properties in the intermediate tasks.

Given the sequential decision process of these systems, these gains significantly improved our system evaluated in our real-world scenario. This curriculum of knowledge transfer not only improves TODS in a general sense but is extremely helpful in contexts where data acquisition is costly, and there are few available examples.

1.5 Thesis Structure

Chapter 2 provides a background introduction to the literature addressed in this investigation. We start by exploring fundamental concepts in deep learning and NLP. Then, we explore studies on curriculum learning, the main foundation of our research. We present studies on transfer learning, especially for the sequential case, where we extended the source-target task framework. Finally, we explore the literature on TODS, the application we found results in, and friendly abstractions to illustrate our methodology.

Chapter 3 presents a synthesis of related studies in curriculum, transfer, and multi-task learning literature. We detail each study and present how it is compared with our present proposal. We discuss investigations in the TODS literature and studies presenting a similar proposal to ours, e.g., computer vision and Reinforcement Learning (RL). To the best of our knowledge, no literature work designed intermediate tasks to improve learning transferability.

Chapter 4 presents and formalizes our methodology. The CTL approach combines curriculum learning and sequential transfer learning for a gradual adaptation to the final task employing intermediate tasks in the learning process. We present a model for formal analysis of encoded-like tasks and how to derive intermediate datasets from them. We present a generalized intuition on the CTL approach and apply it to the TODS problem.

Chapter 5 describes the conducted experiments. We isolate and compare our approach to other known methodologies. We instantiate a task curriculum for TODS by extracting forum data. We evaluate the CTL in direct optimization capabilities and the high-level quality of the agent. In our conducted experiments, we measured the results in the MultiWoZ dataset. We present detailed experimental analyses of language inference and explain the intuition for the generalization provided by the curriculum.

Chapter 6 reports on our exploration to fully implement the curriculum for a "pizza ordering agent" real-world setting. We start from the data creation process and finally implement an end-to-end agent in a Web interface. We briefly present our novel methodology for creating 4-level chatbot agents based on the Wizard-of-Oz approach. We evaluate how our pre-training approach helps to learn when available training examples are scarce.

Finally, Chapter 7 summarizes our contributions regarding the construction of novel datasets, methodologies, and experimental results. We provide discussions on transformer abilities, and generalization from our advances reached. We further discuss the findings and propose future research directions on the open problems observed.

Chapter 2

Background

This Chapter presents the background literature relevant to understand our proposal. We introduce fundamental concepts in NLP and Deep Learning, then present essential concepts central to our proposed solutions.

2.1 Fundamental Concepts

Pre-trained models are the standard approach for inference in complex data (Ruder, 2019). First, we optimize a model to realize some task in a dataset with several examples; later, we adapt this trained model to our final task. **Fine-tuning** technique utilizes a pre-trained model, a network that considers its parameters adjusted to another source task as the initialization weights for the objective task. The weights are slightly modified to adjust for the target task (Howard and Ruder, 2018). With enough labeled data in the initial task, the neural network can acquire good biases as a starting point for the final task.

In NLP, a strong approach for a pre-training task is language modeling. **Language Modeling** consists in estimating the conditional probability of an element in the language given the previous context. For example, in a word form, we estimate $P(w_n|w_{n-1}, w_{n-2}, ...)$. For neural networks, we approach this by minimizing

$$\mathcal{L}_{\text{ML}} = -\sum_{n} \log p_{\theta}(w_n|w_{<n}).$$

When neural models estimate conditional probabilities, they learn deep language patterns suitable for several tasks in NLP (Brown et al., 2020).

When dealing with variable-length input-output sentences, representing the running information in a fixed-size hidden state allows better semantic derivations of the model. **Encoder-Decoder** architectures consist of the following: (i) an encoder model that compresses the input sentence in an intermediate fixed-length representation; and (ii) a decoder model that uses this compressed information to reconstruct the output sentence (Cho et al., 2014). In this process, we force the model to learn a simplified deep representation of the running data.

The **Attention Mechanism** improves intermediate representations by allowing variable intermediate states to attend to farther points, so the model could search for important semantics in the entire sentence and construct meaningful relations between the sequential input (Bahdanau et al., 2015). Later, Vaswani et al. (2017) proposed to create a model entirely based on attention blocks, named **Transformer**. Instead of using recurrent architectures, which suffer from vanishing gradients and optimization problems, they used a full attention model composed of an encoder-decoder that can attend to past or forward words even wider than previous architectures.

GPT-2 uses a causal approach for language modeling (Radford et al., 2019) to exploit only the Decoder block of the transformer. The **Causal Language Modeling** approach predicts the n-th token based on the previous $n-1$ tokens sequentially. For efficiency, in causal language modeling, all tokens are provided simultaneously. Still, each token in the sequence has no access to the last tokens, so it could not use this information to predict, enabling parallel training without overfitting. This model trained for causal language modeling could be used auto-regressively for **Natural Language Generation**. Given a predefined context, a language generation task synthesizes a sequence of words with contextual meaning and correct grammatical construction. LMs, through context-sensitive auto-regression, could infer sequences of texts with a meaningful response by maximizing the likelihood of the desired task.

When scaled, causal LMs starts to learn many non-trivial properties of language, acquiring the ability to few-shot in a myriad NLP tasks. The **Few-Shot Learning** setting asks a model to predict some unknown task given a few examples. Like humans do by demonstration, the model should only guess the answer given its prompted text, without any gradient updates (Brown et al., 2020; Wang et al., 2020).

While LMs are extremely useful in NLP, and specifically in conversational environments, training deep models is not an easy task, as this dissertation addresses. The following sections present concepts explored in this investigation that allow improving the training of LMs, and we explore using them as conversational agents that help customers solve tasks.

2.2 Curriculum Learning

The history of AI is heavily inspired by human learning. Naturally, several pedagogical principles are inherited from this literature. First, exploited in animal training, by the name of shaping (Skinner, 1958), the idea of presenting more complex tasks gradually positively impacts learning. Designing a curriculum to teach humans demonstrated many benefits in the learning process (Dansereau, 1978). By presenting a curriculum, the learners can focus on acquiring basic skills previously and then use these skills to learn more advanced concepts. In the case of language acquisition, posing problems as tasks and order by their complexity demonstrates a huge improvement in language teaching (Nunan, 1991).

This approach for training NNs could be interpreted as a continuation method in optimization literature (Bengio et al., 2009). Remarking that complex instances have nosier optimization surfaces, more simple examples are easier to find the global optimum and somehow could describe the global problem that is attempting to be optimized.

Considering auto-regressive language modeling sentences from corpora with cross-entropy as cost functions $C_\lambda(\theta)$, where C_0 is easily optimized, and C_1 is our target. We can optimize for $C_0(\theta)$ and gradually increase λ constraining θ to the minimum of $C_\lambda(\theta)$. As in Bengio et al. (2009), we can sort (or schedule) the training examples given some complexity measure $\mathcal{H}$; usually entropy, such that for a given distribution of sentences Q_λ, we have:

$$\mathcal{H}(Q_\lambda) < \mathcal{H}(Q_{\lambda+\epsilon}) \qquad \forall \epsilon > 0 \tag{2.1}$$

and the sorting weight for each sentence z is given by

$$W_\lambda(z) \leq W_{\lambda+\epsilon}(z) \qquad \forall z, \forall \epsilon > 0. \tag{2.2}$$

Elman Elman (1993) was the first to address the problem of gradually learning more advanced grammar in NNs. His work used a char-level Recurrent NN, and later formalized by Bengio *et al.* Bengio et al. (2009) by the name of curriculum learning. Many studies later explored this curriculum approach Soviany et al. (2022). In RL literature, we have examples proposing the creation of intermediate tasks for smoothing learning the final task Narvekar et al. (2016); Foglino and Leonetti (2019). Examples in computer vision demonstrate intermediate tasks introduced to help the learner acquire abstractions for the final task Pentina et al. (2015); Soviany et al. (2021).

Improving the initialization point and convergence path is a general procedure for better optimizing models. The next section presents the transfer learning method for better convergence.

2.3 Transfer Learning

Using additional data for learning a target task is a central aspect of modern deep learning, as it helps improve the model bias. There are two main approaches for transfer learning, multi-task learning, where the additional tasks help learn some aspects of the target task, and sequential transfer learning, where the model is trained on a task and later fine-tuned in another task. The first task also helps in the target task but is more general than the target task.

The first study to propose sequential transfer learning dates from 1976 Bozinovski and Fulgosi (1976). In the context of NLP, it is pretty recent. Radford *et al.* Radford et al. (2017) proposed to predict the next word with an LSTM model, which presented a good initialization for sentiment analysis. Then, ULMFiT from Howard and Ruder Howard and Ruder (2018) proposed to use language modeling as the source task and fine-tune it to various tasks with successful results. Since then, the standard approach for almost any task in NLP has been to fine-tune a huge pre-trained LM in the specific target task to archive good results.

Using the enormous generalization ability that pre-trained LMs had to other NLP tasks, Radford *et al.* proposed GPT-2 Radford et al. (2019). It uses the decoder of a transformer architecture to train an auto-regressive causal LM. This architecture learned next-word prediction on a vast data set. Several language-generation tasks, like text summarization and response generation, were improved by fine-tuning this pre-trained model. BERT Devlin et al. (2019), unlike GPT, uses an Masked LM applied to the encoder part of the transformer and used the initial sequence token (<CLS>) to perform a classification task. BERT pre-training consists of a multi-task objective. The pre-trained BERT got state-of-the-art results in natural language inference, named entity recognition, and question-answering tasks.

T5 approached multi-task learning by presenting all problems as a text-to-text inference Raffel et al. (2020), where the prefix of the input determines which task the model should perform. For example, to translate a sentence from English to German, we present the following input to the model: "Translate English to German: That is good". With this setting, the authors of T5 applied the original transformer architecture to several NLP tasks without any modification, allowing massive training on many simultaneous tasks that allowed T5 to score on top of many benchmarks.

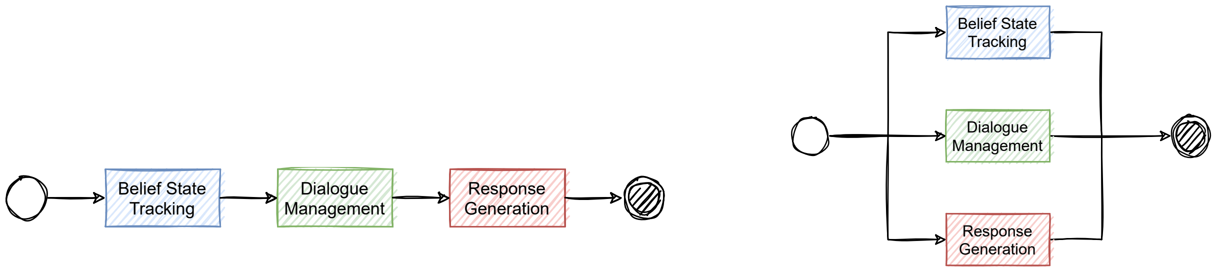
Those models present an exciting property. Researchers executed other out-of-domain tasks with few or no training data examples with reasonable precision. A great example of this capacity is the GPT-3 model Brown et al. (2020), a decoder architecture with 175 billion parameters. Without fine-tuning to target data and performing a few-shot with no gradient updates, it could improve over previous state-of-the-art models in more than 10 points in accuracy for question-answering tasks, where previous models had an accuracy with no difference of random chance Hendrycks et al. (2021a). Another observed phenomenon for these large models was scaling laws. As we increase the model size, dataset size, and amount of computation used for training, the effectiveness improves continuously by a power-law Kaplan et al. (2020); Henighan et al. (2020).

These large models' great generalization, transferability, and scaling properties boosted the results in several NLP applications. In the following, we present the advantages of LM in TODS.

2.4 Task-Oriented Dialog Systems

TODS are grouped in two methodologies: pipeline architectures and end-to-end architectures Zhang et al. (2020d). Pipelines manage the dialog through a series of components and decision steps that, in the end, produces actions and responses for the given utterance and context. These components commonly are Belief State Tracking, Policy Optimization, and Response Generation (cf. Figure 2.1a). On the order hand, the end-to-end architecture has a single model that performs these three functionalities internally (cf. Figure 2.1b).

Dialog State Tracking (DST) aims at each step to estimate the intents given last and previous user's messages. These intents are modeled similarly to a classification task in which each utterance belongs to a known business issue. In a second level, each business intent has information slots that should be filled to complete a task Mrkšić et al. (2016). Similar to when you ask for a restaurant, the price range and the type of food are



(a) Pipeline Architecture. (b) End-to-End Architecture.

Figure 2.1: Taxonomy for Task-Oriented dialog systems organization.

examples of information that should be filled in to return restaurant options. In Zhang et al. (2020d), they classify slot filling as free-form or fixed vocabulary; free-form consists of predicting the intent given a selected range in the user's message, for fixed vocabulary aims to classify a message given a previous vocabulary.

Dialog Policy (DP) is similar to the concept in reinforcement learning literature. It aims to decide which action the system should take given the actual state and belief Zhang et al. (2019). Several techniques are applied to design and optimize policies, supervised learning, simulations, and online reinforcement learning. Given the high costs of real-world data, studies have addressed this problem through user simulation dialogues Schatzmann et al. (2007); Li et al. (2016); Shi et al. (2019).

The Response Generation (RG) module comes in two forms: response selection, given a list of predefined responses, should return the more suitable response to the user. The second method, Natural Language Generation (NLG), synthesizes a unique response for this combination of knowledge, given the question, context, and previous experience. The NLG approach often applies a LM trained on large corpora to induce a conditioned sequential response. Although the NLG alternative prompts a better experience for the user Foster (2019), it needs large amounts of textual data for high-quality response generation Radford et al. (2019); in specific contexts, like oriented dialogues, this data is a scarce resource Huang et al. (2019).

Pipeline methods produce complex models that need many engineering solutions to address business specificities. As each module operates standalone, an improvement in a single module does not necessarily translate into a better improvement for the whole system Gao et al. (2018). In online environments, each module's incorrect inference propagates the error to subsequent ones Liu and Lane (2018).

The end-to-end architecture, inspired by multi-tasking NLP approaches (Hosseini-Asl et al., 2020), applies a single architecture to solve the TODS task. As input, it first gets the user utterance (and previous context) and models it as a sequence-to-sequence problem. The output is a sequence containing the belief state, system action, and generated response, often done by maximizing the likelihood of previous dialog examples. Compared to the pipeline method, the end-to-end can handle long dialogues through modern Transformers implementations Tay et al. (2021). Language modeling has, in general, better generalization Raffel et al. (2020). The last aspect to notice is that as the entire task is encoded in the output sequence, the optimization of this modeling advances the whole system's operation.

The first approach to solve TODS through end-to-end modeling was proposed by Wen *et al.* Wen et al. (2017). Their work modeled the problem as a sequence-to-sequence prediction and delexicalized the text to reduce language variability. Compared to previous models, this modeling enables the architecture to generalize to any domain without needing handcrafted modules. This network architecture uses RNNs for encoding a latent representation of the dialog; and has many blocks, each dealing with a part of the dialog. Later, other proposals improved on these basic ideas. For instance: memory networks Bordes et al. (2017); multi-task learning and reinforcement learning Liu and Lane (2018), and multi-hop attention mechanisms Madotto et al. (2018).

Schuster *et al.* Schuster et al. (2019) proposed using attention mechanisms as part of the architecture and cross-language techniques for transfer learning. Zhang *et al.* Zhang et al. (2020b) suggested a technique of data augmented based on recurring states of the dialog. It uses attention mechanisms using pre-trained word embedding as initialization. Zhang *et al.* Zhang et al. (2020c) proposed a pre-trained model in Reddit and Twitter conversations as a response generation alternative.

Peng (2020) Peng et al. (2020) used a pre-trained GPT-2 to decode the objective by predicting the belief state, database state, and response. Hosseini *et al.* Hosseini-Asl et al. (2020) developed the SimpleTOD, improving the previous work by adding a system action to the decode objective. This encapsulated the states by a token, significantly improving TODS in the MultiWoZ dataset. Based on SimpleTOD, Yang *et al.* Yang et al. (2020) trained the GPT-2 at a session level instead of the turn level used in SimpleTOD, improving the overall accuracy of the UBAR. SUMBT+LaRL Lee et al. (2020) and LAVA Lubis et al. (2020) applied reinforcement learning for the response generation step. DoTS Jeon and Lee (2021) used state tracking and only the current context to reduce memory consumption and maintain good results. Liu *et al.* Liu et al. (2021a) used a clever decoding technique that generates more specific responses.

2.5 Synthesis

In this Chapter, we presented the building blocks and key concepts that our investigation relies on. We first explored the curriculum learning literature, which consists in sorting instances of training examples concerning some measure of complexity. This approach allows a smoother version of the original problem to be optimized earlier. Later, the harder ones are optimized with a better starting point not to be stuck in local optima and arrive at global optima.

The transfer learning literature focuses on using auxiliary learning. Sometimes using more information from the same target task or another completely different task helps acquire the right bias for the target task. In the case of NLP, most studies adopt language modeling as the source task. By the distribution hypothesis, good representations are learned for every word in the vocabulary, so the target task could use this rich information and quickly adapt to a new task.

We introduced current approaches in the field of TODS describing belief tracking for identifying user intent, dialog policy (management) to choose which action to take, and response generation to return textual feedback. State-of-the-art methods use end-to-end architectures to reason over all components for better overall effectiveness. Lately, they encoded the problem in a sequence-to-sequence task to enable LMs to solve this using its boosted NLP skills.

In the next chapter, we explore related work to discuss and understand the latest studies conducted in the literature that resembles our proposal. We review studies to demonstrate how our work differs from existing literature.

Chapter 3

Related Work

In this chapter, we discuss transfer learning and curriculum learning components in the context of LM, TODS, Computer Vision, and RL. We describe how using knowledge from a different distribution can help the learning process; and how ordering examples can produce a smoother optimization. We relate how curriculum and transfer learning are a prologue to our solution.

3.1 Curriculum Learning

In NLP, curriculum learning was initially proposed as a criterion for sorting sentences given their vocabulary frequency (Bengio et al., 2009). It started with a window of 5000 most frequent words, and sentences that have words out of this window were discarded. The vocabulary is expanded with more than 5000 words, and the same process is repeated until the entire vocabulary is covered. This procedure allowed the LM to optimize for a better loss with much more stability than random ordering samples.

Many studies are exploring curriculum in an instance-level (Wang et al., 2021), where each example presented to the learner respects a complexity order. For a task-level curriculum, fewer studies applied it to improve learning. Pentina et al. (2015) extended the concept of curriculum for a task level, with a dataset of many feminine shoes; each pair of shoes having many attributes, the curriculum was sorted based on the intuitive difficulty of classifying the attribute. It first started by classifying if the shoes were shiny and continuously evolved for harder classifications, like if the shoes had high heels, and later if it was formal.

We found a few examples of studies exploring curriculum at a task level. In the RL literature, which is heavily inspired by pedagogical principles, authors have explored the usage of curriculum for learning agents. Narvekar et al. (2016) applied a curriculum for learning to play chess. They got the data from the same generating process and posed easier to hard versions of QuickChess. With this setting, the loss curve was better in its trajectory compared to no curriculum.

Fang et al. (2019) used a curriculum for hand manipulation, where the robot's final achievement was to manipulate a pen; as an intermediate task, it learned to handle easier objects, like blocks and eggs. Foglino and Leonetti (2019) increased the environment's complexity by extending steps, adding more elements, and changing the environment objectively. Both approaches showed that the agent does not converge in some cases without a curriculum, as the optimization surface is too noisy and it gets stuck in local optima. Another relevant study that used curriculum at a task level was conducted by Liu et al. (2020). The authors proposed adjusting the task gradually, transforming the auto-regressive translation to a non-auto-regressive task using the same dataset.

In most recent studies for TODS and NLP in general, the curriculum is applied in an instance-level (Soviany et al., 2021; Wang et al., 2021) to improve model convergence and global optima. Most of them are based on pre-trained LMs. For example, Kim et al. (2021) used a RoBERTa and BART model to predict slot values and generate responses, respectively. Their study employed a teacher model to score examples (using F1-score and BLEU) from easier to harder. This ordering was used for fine-tuning both models.

Another example of an instance-based curriculum was proposed by Dai et al. (2021), which defined a curriculum as the combination of both model difficulty, cross-entropy, and human-selected metrics, following the "baby steps" scheduling method. As it explored previous knowledge of the schema, this approach improved DST performance by a meaningful margin.

A set of studies focused on using the teacher-student setting to guide the curriculum. Zhu et al. (2021b) combined knowledge distillation and curriculum learning for training an agent. They used response length, utterance entropy, and coherence as the complexity measure for scheduling a student's training over the teacher. They evaluated the method on open-dialog data where the task consists only of NLG.

Several studies in TODS used a curriculum to control environment complexity when the problem is set as a RL task. Saito (2018) used a curriculum based on the number of slots to improve the slot-filling ability. At each new episode, they allowed the model to train on examples that required more slots. Liu et al. (2021b) used a curriculum to sort dialogues and automatically computed reward information to measure environment complexity. This approach boosted the effectiveness of DP learning by a reasonable margin.

Zhu et al. (2021a) trained a Deep Q-Network agent where the teacher controlled the environment complexity through a curriculum measuring sizes of informed and required slots. This approach allowed the learning agent to acquire positive rewards faster than other compared methods. Zhao et al. (2022) used the return reward from trajectories in an experience replay grouped by user goal to compute the sample complexity. The sample complexities were later used to schedule the adaptive training, and this approach improved DP as it presented more stability in the learning curve.

Table 3.1 shows how our study compares to related literature. Most studies focus on using curriculum learning at an instance level, improving the optimization process's convergence. However, they do not present meaningful gradual examples, as the same grammar generates them. Studies based on a curriculum by task better resemble our approach. As a distinct process generates data, we have two advantages: obtain data from a cheaper source, and pre-trained data allows a more gradual adaptation. Our proposal addresses an even more profound aspect of the target task because we directly design easier examples for the target task. These easier examples are not restricted to costly data acquisition, *e.g.*, our costs were marginally none.

Table 3.1: Comparative table for Curriculum Learning literature

Work	by Instance	by Task	by Grammar
(Bengio et al., 2009)	x		
(Pentina et al., 2015)		x	
(Narvekar et al., 2016)		x	
(Saito, 2018)	x		
(Fang et al., 2019)		x	
(Foglino and Leonetti, 2019)		x	
(Liu et al., 2020)		x	
(Liu et al., 2021b)	x		
(Kim et al., 2021)	x		
(Dai et al., 2021)	x		
(Zhu et al., 2021b)	x		
(Zhao et al., 2022)	x		
CTL (ours)		x	x

3.2 Sequential Transfer Learning

In the previous section, we presented a series of studies where most of them were based on pre-trained LMs on general text, and later they were fine-tuning following a curriculum. Most modern approaches in general for TODS applied pre-trained LMs, for DST, RG, DP or end-to-end, like GPT, T5, BART, or BERT. This section presents investigations that non-trivially explore transfer learning to improve these systems' effectiveness.

In modern NLP, researchers have transferred knowledge of large deep neural networks in multiple contexts. A single model is trained once and then adapted to multiple tasks with varying goals, such as question answering or natural language inference Devlin et al. (2019). This process is called sequential transfer learning. This transfer learning approach aims to train a model on one source task and then adopt the model for another final task Ruder (2019).

Sequential transfer learning consists of two steps: 1) Pre-training: A model is trained for the source task on a large dataset with high computational costs and long training times. A common goal of the source task should be very generic and applicable to many target tasks; 2) Fine-tuning: The pre-trained model is transferred to a downstream (target) task with much less data. This phase has a shorter period, and the target tasks can benefit from learned representations from a primarily pre-trained source model.

The literature presents a set of studies that explore “conversation-biased” data for improving the pre-trained capabilities of LMs. For example, Wu et al. (2020) proposed pre-trains a BERT model in several TODS datasets, acquiring a considerable improvement in recognizing entities, intents, and response selection compared to the standard approach, it is, LMs trained on general text corpora.

Zhang et al. (2020c) proposed pre-trained a GPT-2 architecture in an unsupervised Reddit corpus in which a conversational bias was presented. With this approach, human evaluators often preferred DIALOGPT answers compared to strong baselines for open-domain response generation.

In Adiwardana et al. (2020), the Meena model was introduced for open-domain multi-turn conversation. Its main appeal is an unprecedented scale for this kind of model; they trained a 2.6B parameter model on context-response pairs extracted from social media with 40B words. With the large-scale effects, Meena had a smaller gap between machine-generated and human-generated responses compared to other chatbots.

The Thoppilan et al. (2022) work extended Meena by pre-training a super-massive model near GPT-3 scale (137B). This model was trained on conversational data with a simple utterance-response setting. In addition to the unsupervised pre-train, the authors fine-tuned it on annotated data and allowed the model to query external sources to improve safety and factual grounding. With this approach, the model acquired human-like performance measured by the SSI metric.

For the computer vision domain, Liu and Ji (2021) applied multiple pre-training steps to learn to classify images. For each pre-training step, the authors explored a differently labeled dataset, starting with general image labeling, then training on medical images with many available examples. Finally, they fine-tuned to target data where low examples were available.

In Table 3.2, we present how our work relates to other works in transfer learning literature. Many works explored corpora with strong conversational bias. Some of them presented it in an utterance-response setting; however, none explored particular encoding to match the grammar of the target task.

Table 3.2: Comparative table on how works explore transfer learning for pre-training conversational models.

Work	Conversational Bias	Utterance Response Bias
Wu et al. (2020)	x	
Zhang et al. (2020c)	x	
Adiwardana et al. (2020)	x	
Thoppilan et al. (2022)	x	
CTL (ours)	x	x

3.3 Multi-task Learning

In the end-to-end TODS, recent studies focus on architectures for rapid adaptation or learning with low resources. In Zhang et al. (2020a), the authors proposed applying semi-supervised learning in a latent representation of the belief state through variational inference. This could assign pseudo-labels for unlabeled examples, especially for the MultiWoZ dataset (Budzianowski et al., 2018) only needs half the data set to be annotated to get the same accuracy.

In Lin et al. (2020), instead of feeding the model with the complete running conversation, they managed a copy mechanism that is responsible for just editing the last belief state and encoding in a latent representation named Levenshtein belief spans. This latent representation improves dialog state tracking and boosts transferability between domains. With only 20% of data, they matched the performance of strong baselines trained on full MultiWoZ.

Peng et al. (2020) encoded the entire TODS task into a single concatenated sequence, similar to Hosseini-Asl et al. (2020); Yang et al. (2020). However, the authors used end tokens to segment tasks instead of beginning and ending tokens. In addition, the authors applied multi-task learning in the sequence-to-sequence problem to a contrastive objective. Each token also should classify as matching or mismatching the current token.

In Kulhánek et al. (2021), in addition to the sequence-to-sequence prediction, they proposed an auxiliary task for the auto-regressive decoder. The last decoded token also had to predict the inconsistencies similar to SOLOIST but used a more efficient corruption strategy. This multi-task learning approach improved the model's ability to generate consistent answers.

Su et al. (2021) used multi-task learning to improve the model's few-shot capabilities. Instead of applying multiple optimization objectives, their work explored the sequence-to-sequence multi-task setting as in T5. Another critical factor is that the authors trained the model on multiple TODS data sets, like MetaLWOZ, CamRest676, and Taskmaster.

Lee (2021) proposed to multi-task learning both span based annotation, through NER and the sequence-to-sequence TODS task. Empirically, the authors demonstrated that this auxiliary task improved the model's ability to recognize belief states.

Cholakov and Kolev (2022) proposed to use an auxiliary task to improve various metrics in MultiWoZ. The network uses a T5 architecture for sequence-to-sequence solving the task and incorporates a binary classifier to select the correct response between distractive responses, so it is trained with two objectives simultaneously.

In Sun et al. (2022), the authors used the Levenshtein belief to reduce inference time. They proposed a denoising objective to reduce error propagation and boost belief tracking and response generation. The denoising objective is made by two objectives, back reconstruction, which reconstructs the current dialog state from a latent representation, and denoising reconstruction, where it has to predict masked tokens from the input.

He et al. (2022) added positional embedding for the role and turn tokens, included a pre-training objective to semi-supervised data, and implemented a multi-task objective where they had to predict the dialog act, response selection, minimize the consistency regularization, and generate a response.

Table 3.3 demonstrates how studies in the literature relate to our work. We filtered those that better could be related to ours. Generally, studies explored multi-objective optimization, like editing distance plus binary cross-entropy or additional annotations in data. For instance, they join TODS data sets created by other groups (MultiWoZ, Frames, etc.).

3.4 Synthesis

We found that in the curriculum learning literature, no work employs a curriculum at a dataset level for NLP. In RL and Computer Vision, some studies increased the difficulty

Table 3.3: Comparative table on how works explore multi-task learning to improve model prediction in TODS.

Work	Multi-Task Optimization	Sequence-to-Sequence
Peng et al. (2020)	x	-
Kulhánek et al. (2021)	x	-
Su et al. (2021)	-	Additional Data
Lee (2021)	x	-
Cholakov and Kolev (2022)	x	-
Sun et al. (2022)	x	-
He et al. (2022)	x	Pre-trained
CTL (ours)	-	Intermediate Tasks

of the environment to be solved as a means to smooth the target objective.

In the sequential transfer learning literature, earlier attempts employed pre-training on strong conversational bias instead of default general language. Other more recent studies encoded open-domain conversation in a simple question-answer setting. We showed no work explored pre-training tasks directly addressing the grammar in the final TODS task.

In fields like RL and Computer Vision, the authors used multiple steps of transference but without any formal definition for the ordering. To the best of our knowledge, no literature for TODS uses a sequence of data sets for gradual adaptation.

For the multi-task learning literature, several studies explored extra annotation. Additional annotated data that follows the same generating process or optimizing for multiple objectives. This dependence on over-annotated data does not transfer to other languages, as they depend on English datasets or do not generalize for domains out of scope.

The next chapter presents our solution for reducing data dependency in a language-agnostic methodology.

Chapter 4

Grammar Acquisition with Curricular Transfer Learning

Our proposed CTL methodology aims to reduce the requirements for annotated data in training sequence-to-sequence tasks. We manipulate the structure of existing abundant data, by exploring the present structure in web-data, to create intermediate tasks to help the target task, which we call “data hacking”. We create data that simulates a simplified version of the grammar, so our model can acquire properties of this grammar and later generalize to more complex ones. In this chapter, we present a general theory for complex grammar acquisition by grammar decomposition and instantiate the solution for the problem of TODS.

4.1 Framing the Problem

The classical transfer learning framework, using sequential induction, allows us to use out-of-distribution data from a source task to leverage the knowledge in this data for another correlated target task. By learning the proper bias for the problem to be modeled, the learner model can rapidly adapt to the target hypothesis, demanding fewer examples to acquire a good generalization point in the target domain. In computer vision, transfer learning is mainly conducted by classification in massive labeled image/video datasets. In the context of NLP, transfer learning is achieved by pre-training an LM, with self-supervision, in general texts sampled from indexed pages on the web (Devlin et al., 2019; Radford et al., 2019; Raffel et al., 2020), generally with miscellaneous and diverse contents.

Although evidence shows that language modeling acquires high generalization capabilities for the out-of-distribution data (Hendrycks et al., 2020), some specialized and complex tasks that diverge from the general distribution of random web text are less benefited (Thoppilan et al., 2022). In low-resource contexts, those pre-trained models tend to overfit or underfit the target task, failing to arrive at a desirable optimum and presenting poor generalization. Figure 4.1 presents a pre-trained LM failing to arrive in global optima because it could not seek a good optimization path described by the dashed line. The lack of data conditions to the wrong path, a continuous line.

Some text sequences have a non-trivial structure to learn (Mielke et al., 2019; Cotterell

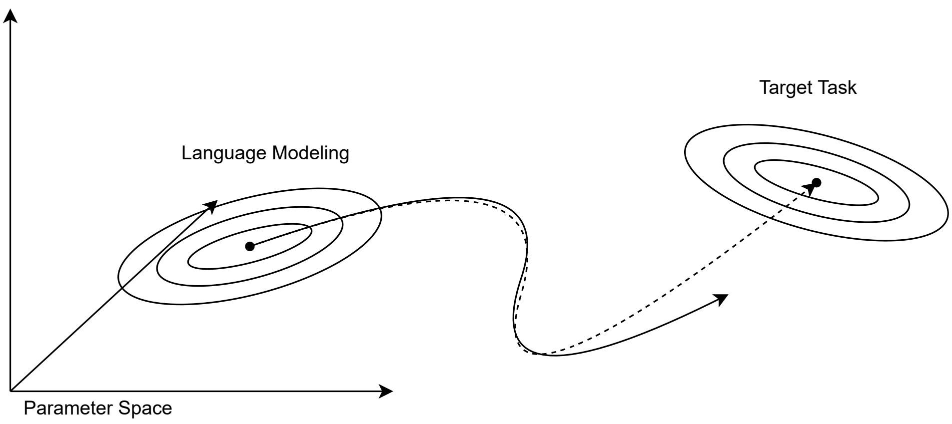


Figure 4.1: Learning a target task from language modeling. With insufficient data, the learning process fails to arrive at a global optimum for the target task.

et al., 2018; Zellers et al., 2019; Delétang et al., 2022), or have a distribution of symbols that is context-dependent (Zhang et al., 2020c; Thoppilan et al., 2022; Lewkowycz et al., 2022; Cobbe et al., 2021). In the case of TODS, the sequence the task aims to learn has a conversation-biased distribution of words, many unique marking tokens for the classification, named entity recognition, and parsing. This drift between pre-training and fine-tuning could be sub-optimal. Exploring other pre-training data that resemble the same properties present in the TODS task could significantly improve the learner.

While LMs parameters are overgrowing, motivated by Moore's and scaling (Hutter, 2021; Hernandez et al., 2021; Bahri et al., 2021; Henighan et al., 2020; Kaplan et al., 2020) laws, the same does not happen for data annotation. The cost of labeling data does not drop every 18 months. And the more complex the dataset, the more costly it is to annotate it (Hendrycks et al., 2021b). Given a fixed budget, fewer examples from this complex data are usually available. We illustrate this observation in Figure 4.2.

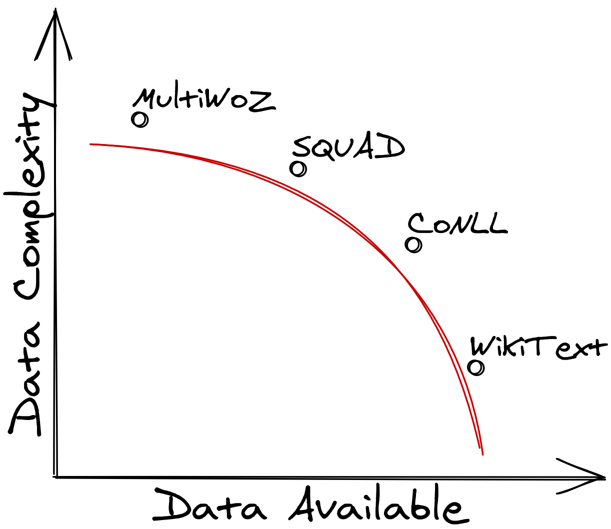


Figure 4.2: Hypothetical supply curve for data complexity versus availability in public datasets. We observe that datasets with higher complexity are more expensive to obtain. In general, we have fewer examples to teach a learning agent.

An alternative to costly datasets is the construction of pseudo-labeled ones, silver-standard datasets. In these solutions, data are artificially labeled by an automatic procedure (Jaderberg et al., 2014), reducing the overall cost of annotation. Although it has derisive costs to build, the level of noise and inaccuracy in such fabricated data could not benefit the learner, as the generalization from an induced learner tends to decrease.

Meta-learning is a capability that LMs have which enables them to learn many tasks indirectly and without any specification. By exploring this ability, we can use naturally annotated data as additional pre-training steps for a more gradual fit to the target task. In this approach, we have a few available examples (cf. Figure 4.3).

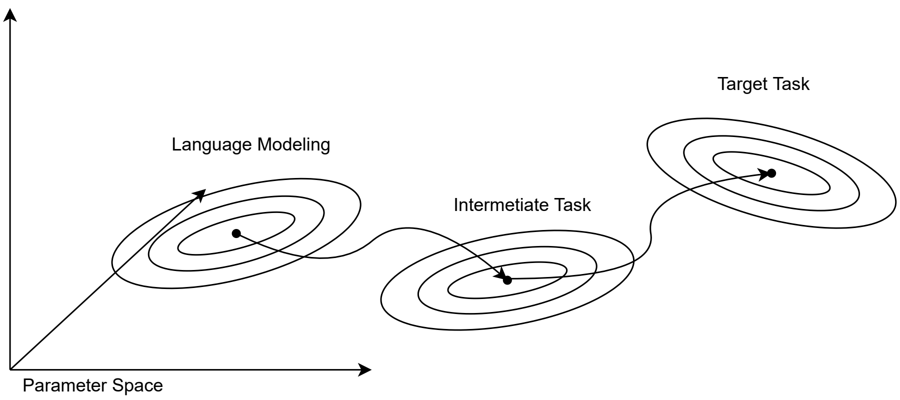


Figure 4.3: Learning a target task from gradual adaptation. With a better initialization, the model needs less data to arrive at the global optima.

4.2 Approaching the Solution

We can view the problem of training a model from two perspectives: 1) the learning perspective, where we aim to obtain the model to acquire the skills to recognize patterns in data; 2) the optimization aspect, where we aim to reach a model to minimize some loss function. As posed by (Bengio et al., 2009), a curriculum approach has a beautiful interpretation for both perspectives. We teach a model in the same way humans learn by gradually increasing the task's difficulty. Relying on this curriculum approach, we might minimize a less noisy version of the original problem to arrive at the global optima, a continuation method.

In the language acquisition process, the generative grammar theory assumes that a learner has an innate universal grammar that restricts what kind of grammar a learner could acquire (White, 2003). The process starts by recognizing simple structures in a grammar; instead of memorizing, the learner identifies syntactic structures they encounter and evaluates the feedback in an environment to determine precisely the grammar being used (Guasti, 2017).

In the early stages, the learner only recognizes simple grammatical structures; the initial process is crucial for complete grammar acquisition, as many possible candidate grammars are filtered in this search step (Komarova et al., 2001). This behavior is explored in formal education. Whereas most of a child's primary language is obtained culturally, without directed control, the process is almost entirely performed in a controlled environment for second language acquisition. In our developed solution, we first present the learner with basic phrases with simple composition, to later extend for more complex sentences (Davies, 1980; Guasti, 2017).

Several approaches in AI explore the advantages of posing easier instances of the problem or addressing sequentially surrogate objectives to achieve more complex goals. In RL, robotic hand manipulation is an example. Instead of directly training the robot to put a red box over a blue box, it's easier first to teach the arm to recognize which color to pick, then how to hold the box correctly, and later how to place it over the blue box Manela and Biess (2022). This decomposition of complexity allows agents to learn the final task faster than directly attempting to perform the final task.

The optimization we described above could be interpreted from two distinct perspectives. The first is the sequential transfer learning approach, in which we have multiple steps of knowledge transfer. In the second interpretation, we have the curriculum approach. In this approach, we order more complex instances of the problem. Still, we have different tasks encoded as sequences, so the ordering respects the task complexity instead of the instance complexity. For our context, we need to find intermediate tasks that help the learner acquire the skills needed for the final task.

The original curriculum approach stands for organizing the curriculum by example; the TODS can organize it at a task level. We introduce an additional task in the encoded sequence of each step in the curriculum. The problem of TODS, and other related conversational problems, presents many sub-tasks that should be addressed to interact in a dialog. In end-to-end models, the tasks are natural language generation, NER, and parsing. When we conduct training by task level, we are not restricted to the same data-generating process as the original problem, as in transfer learning literature, a more broad task is inserted in the pre-training. So we can explore another source or even fabricate examples with reduced costs to address each task.

4.3 Proposed Solution

Figure 4.4 illustrates our proposal for the task decomposition of TODS in the context of pizza ordering. Through a curriculum of sequential transfer learning, we teach the model to learn concepts essential to the final task being addressed gradually.

The solution starts with a) training a general language model. This model is trained in massive crawled data from the web. In the case of GPT-2, it uses the Common Crawl corpus (Radford et al., 2019). This corpus has a corpus ranging from articles, news, blogs, *etc.*, so many kinds of bias are acquired in this step.

In phase b) the solution trains an intermediate sequence-to-sequence task, artificially constructed by exploring the structure of online forums. In this step, we are simply learning conversational biased text, a structure for the task of TODS, and classifying the discussed topic. We want our model to acquire the abstractions needed for the final task in an intuitive sense.

In phase c) we use comprehensive annotated data to teach the model more complex tasks, like named entity recognition and parsing. This could be performed by manually annotating data for general purposes or bootstrapping from a specialized annotator model. Based on this approach, we can transfer the knowledge from the specialized model to our end-to-end model.

In phase d), we train the model for business-specific usage. For example, in step c), we train a model for general pizza delivery, and in phase d), we specialize this model for unique toppings of a specific brand or how to handle payments for a specific region.

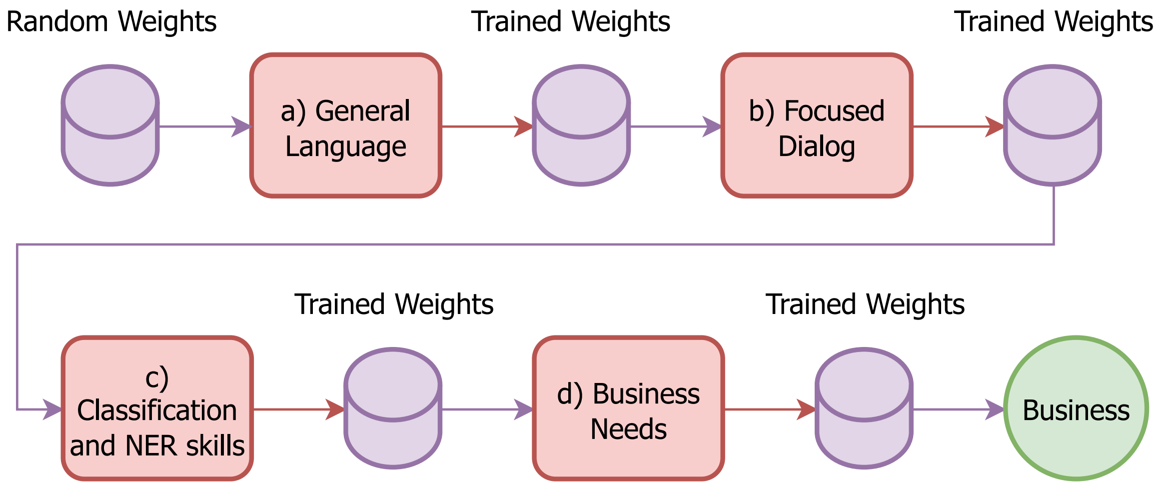


Figure 4.4: Detailed view of a proposed curriculum for conversational environments. From left to right, we start with a general model, and at each step, we gradually adapt the learning agent by retraining on more specialized tasks. This curriculum of tasks allows one to learn complex behavior in context with many examples available and rapidly adapt to domains with low data available.

4.4 Formalization

Here, We informally described our approach. We extend the model of Pan and Yang (2009) for the multi-sequential case presented in this M.Sc. thesis. First, given a domain $\mathcal{D} = \{\mathcal{X}, P(X)\}$ consisting in a feature space $\mathcal{X}$ and a probability distribution $P(X)$, where $X = \{x_1, x_2, ..., x_n\} \in \mathcal{X}$. Consider a task $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$, where $\mathcal{Y}$ is the label space, and $f(\cdot)$ is the objective predictive function, which is not observed but could be learned from the data, where $f : X \to \mathcal{Y}$.

We define transfer learning as, given a source domain $\mathcal{D}_S$, a source task $\mathcal{T}_S$, and target domain $\mathcal{D}_T$ with a target task $\mathcal{T}_T$. Transfer learning aims to help improve the learning of the target predictive function $f(\cdot)_T$ in $\mathcal{D}_T$ using the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$ or $\mathcal{T}_S \neq \mathcal{T}_T$.

Although most studies in NLP literature apply a single step of sequential transfer learning, we propose many steps of transference in the CTL. It is a curriculum of transference, *i.e.*, $\boldsymbol{\mathcal{T}} = \{\mathcal{T}_{S_0}, \mathcal{T}_{S_1}, ..., \mathcal{T}_{S_n}, \mathcal{T}_T\}$, where we have n pre-training source tasks $\mathcal{T}_S$. By this approach, the $\mathcal{T}_{S_k}$ source task helps optimizing for $\mathcal{T}_{S_{k+1}}$ task, acquiring a better generalization for the following task and, consequently, the general curriculum as an accumulation of improvements.

This formulation can be interpreted as a curriculum learning approach. Consider the instances from $\mathcal{T}_{S_k}$ as smoothed objectives of $\mathcal{T}_T$. The cost function C_{S_0} is easier to optimize as it does not compass the entire TODS objective, and we have more available instances in this set. After optimizing for this objective, we gradually increase the task complexity by passing tasks with more linguistic features to extract from C_{S_0} to C_{S_n}; and finally C_T, fewer instances are presented for more complex sets.

Posing the problem in this manner is similar to a continuation method. Continuation methods attempt to minimize a cost function C_0 that resembles some other target cost function C_1 for some family $C_\lambda(\theta)$ of cost functions. We can easily optimize C_0. The target function is a noisier version of that one. In the language modeling problem, the cost function we aim to minimize is the cross-entropy for the next word. The λ family is composed of languages with increasingly complex grammar. For the problem addressed in this investigation, the more complex grammar is composed of simpler grammar learned through language modeling in previous steps. Extending the Bengio *et al.* definition (Bengio et al., 2009), the λ factor varies between instances according to each data set.

In the case of auto-regressive language modeling, the optimization is performed by next token prediction. Suppose a serialized decoding is learned to a specific pattern. In that case, *e.g.*, the regular grammar of starting and ending fields, adapting to a new grammar that shares the same pattern, will be more straightforward. Figure 4.5, presents this observation; the probability of the token to come after <eos_u> is previously learned in a more manageable task, so the optimization in this phase should only focus on learning the classification task.

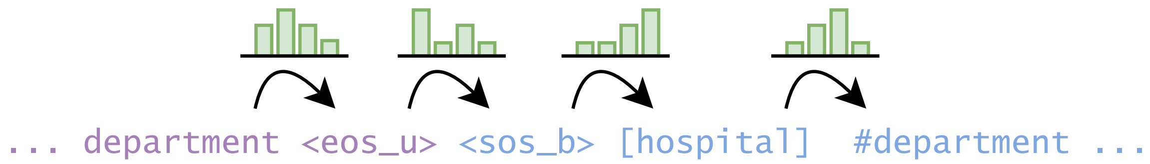


Figure 4.5: Auto-regressively decoding of a sentence in MultiWoZ.

Considering that our final task is composed of formal grammar. Suppose another data source could simulate some derivation nodes in the target grammar without loss in the global structure, *e.g.* the first production rule. In that case, we can pose this simpler version of the grammar as a step in the pre-training curriculum and recursively append more simplified grammars that simulate previous ones.

We formally call it a curricular transfer learning if, given an ordered set of sequence-to-sequence tasks $\boldsymbol{\mathcal{T}} = \{\mathcal{T}_S, \mathcal{T}_{I_1}, ..., \mathcal{T}_{I_n}, \mathcal{T}_T\}$, some complexity ordering $<_\mathcal{C}$, and some grammatical similarity $\sim_G$. Every task in $\boldsymbol{\mathcal{T}}$ respects the order

$$\mathcal{T}_a <_\mathcal{C} \mathcal{T}_b \qquad \forall a, b : a < b \tag{4.1}$$

where a and b are indexes for tasks in $\boldsymbol{\mathcal{T}}$ and

$$\mathcal{T}_a \sim_G \mathcal{T}_b \qquad \forall a, b. \tag{4.2}$$

The main appeal of this method is that more accessible instances of the problem have a derisive cost to obtain, while final instances have a considerable cost. The artificially created intermediate tasks should help the LM to meta-learn the final task, like learning unique tokens to classify, perform a name entity recognition, or generate a response.

We describe the method for complex grammar acquisition in the following manner:

1. Describe the desired grammar formally to be learned, listing the expected types of text distribution and meta-structure.

2. Find text and structures that can be morphed to simulate this behavior. For the same kind of word distribution in the environment (formal, conversational, etc.), some NLP tasks are present in the problem, like classification or NER, etc.

3. Recursively proposes (1) that address the subsequent grammar as another pre-training step.

In the next chapter, we present how we perform these steps for the case of MultiWoZ.

4.5 Synthesis

A general intuition for creating intermediate tasks is to find textual corpora or data sets that we can transform into a form that resembles our final grammar. In the next chapter, we demonstrate how we apply this methodology to improve the optimization and overall effectiveness of MultiWoZ data. We present a simple curriculum decomposition with a meta-structure and a single additional task surpassing previous pre-training approaches.

Chapter 5

Experimental Evaluation

We report on our evaluation to experimentally understand the influences of a curriculum in a learning strategy. Given the substantial computational costs required by all curriculum variations, we suppress the detailed exploration by qualitatively analyzing the resulting models.

We evaluate the CTL in a standard way, using the entire dataset and in a low-resource setting, varying the fraction of data for training. We observe that the loss curve for our approach is always better compared to other pre-training, and overall we obtained improved metrics for `BLEU`, `INFORM`, and `SUCCESS`.

5.1 Objectives

A complete curriculum for TODS might involve several pre-training steps. For this experiment, we analyze a single intermediate training step. We proposed to compare the insertion of a pre-training step that simulates TODS data but is obtained in an unsupervised way. This intermediate task is performed for acquiring regular grammar, and a simple classification task, where we try to classify which topic is being discussed to simulate intent detection. We compare our experiments with a standard language model, and a curriculum with just a conversation distribution, without any particular encoding.

5.2 Data sets

For the conversation-oriented curriculum, we consider a diverse corpus for the first stage of learning to meta-learn general language skills; a corpus that simulates the conversational patterns and distributions of words; and finally, the data set for the target task.

We use the following text data sets as pre-training steps:

1. **Common Crawl (General Language Modeling)** To reduce computational costs for the experiments, we count with HuggingFace[1] for downloading the pre-trained weights for GPT-2 architecture. This dataset was proposed as the unsupervised

[1] `https://huggingface.co/`

auto-regressive pre-train for the proposed architecture Radford et al. (2019). Although we do not train the GPT-2 from scratch in this dataset, we describe it here to illustrate the CTL approach.

2. **Tripadvisor (dialog Modeling)** We built this data set by crawling major cities discussed in Tripadvisor[2] (Paris, Rome, Istanbul, Barcelona, Madrid, Amsterdam, Lisbon, and London), for each thread we encoded the thread creator with user utterance tokens, the city as the belief tokens, and thread replies as system response. We replicate each pseudo-user utterance for each pseudo-system response. The crawler is available in Github[3].

3. **Multi-Domain Wizard-of-Oz (General Task-Oriented Modeling)** The MultiWOZ is a general domain task-oriented dialogues data set collected in a Wizard-of-Oz setting. It comprises 10000 dialogues in domains like 'restaurant,' 'train,' and 'hotel booking.' We selected the latest official data set for our evaluation, version 2.2.

5.3 Architecture

The architecture used for modeling TODS is the autoregressive GPT-2 Radford et al. (2019) based on the decoder module of a Transformer, replicating the same setting in UBAR Yang et al. (2020). Given a sentence of n tokens in $\mathbf{I}$, we assign for each token an embedding vector in $\mathbf{R}^d$, and a sinusoidal positional embedding vector in $\mathbf{R}^d$ given his index. Each embedding and respective positional embedding are summed up and stacked in a matrix $X_0 \in \mathbf{R}^{n \times d}$. Then, we process this X_0 matrix by applying a layer normalization resulting in $\overline{X_0}$ and then applying multi-head attention in the normalized matrix given in Equation 5.1, outputting M_0.

$$
\begin{aligned}
\text{MultiHead}(X, k) &= [h_1; ...; h_k]W_o \\
\text{where} \quad h_j &= \text{Attention}(XW_j^1, XW_j^2, XW_j^3) \\
\text{and} & \\
\text{Attention}(X, Y, Z) &= \text{softmax}\left(\frac{\text{mask}(XY^\top)}{\sqrt{d}}\right) Z
\end{aligned}
\tag{5.1}
$$

Now, we sum the normalized matrix with the transformed matrix $H_0 = \overline{X_0} + M_0$, and apply a layer normalization, getting $\overline{H_0}$, now we apply a feed-forward block with a ReLU activation and sum to $\overline{H_0}$ as a residual, getting $X_1 = \text{FF}(\overline{H_0}) + \overline{H_0}$.

We repeat the above procedure l times until we arrive at the X_l. Finally, we apply the last layer normalization and multiply by an $W^{n \times \text{vocab}}$, given that vocab is the vocabulary size, where we apply a softmax function. Optimizing this with cross-entropy loss, we could maximize the likelihood of predicting the next word given the previous sequence of words. Figure 5.1 presents the representation of this network.

[2] `https://www.tripadvisor.com/`

[3] `https://github.com/jadermcs/tripadvisor-dialogues`

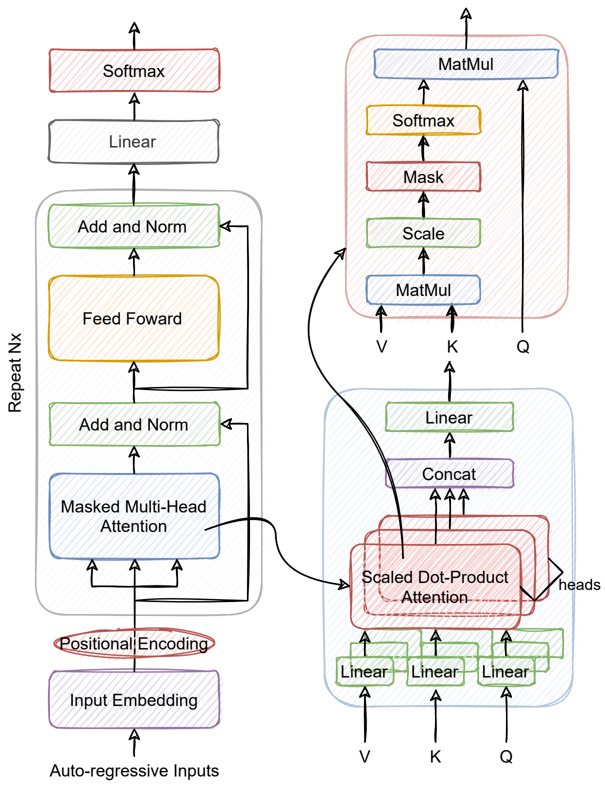


Figure 5.1: GPT-2 architecture (Radford et al., 2019)

5.4 Procedure

To formally illustrate the process of task decomposition, we analyze the case of MultiWoZ encoded as a sequence-to-sequence grammar as in UBAR (Yang et al., 2020). We present the target grammar we aim to learn, how to decompose it into more straightforward tasks, and how to generate intermediate data that resembles the same process as TODS.

We evaluate our methodology under the UBAR setting interacting with MultiWoZ. First, we start with a user prompting an utterance to the system. Then, the system identifies the belief state and queries some external knowledge with this information, then with a response from a database or API, which is turned into an action, it generates a delexicalized response to the user, a response where slot values are replaced with place holders, these placeholders in the response is then replaced with the lexicalized information extracted from the database. Figure 5.2 presents a high-level view of this interaction.

To our LM learn how to solve this problem, we need to encode this problem as a sequence-to-sequence prediction problem. To this end, we use the special `<sos_X>`[4] tokens. The example below represents the first interaction in the dialog as in Figure 5.2, encoded as a sequence-to-sequence problem.

```
<sos_u>I am looking for a cheap restaurant in the center of
the city.<eos_u> <sos_b> [restaurant] pricerange cheap area
centre<eos_b><sos_a> [inform] choice [request] food <eos_a>
<sos_r> There are [value_count] restaurants that meet your
```

[4]where $X \in u, b, a, r$ for the start and end of string in utterance, belief, action, and response

```
criteria. What type of food do you like? <eos_r> <sos_u>...
```

First, we formalize the target grammar to acquire and enlist known properties of this grammar. The training example is the concatenation of the entire dialog session, so the language of this grammar could be any form in $\mathcal{L} = \{\epsilon, UBAR, UBARUBAR, ...\}$. Table 5.1 illustrates the grammar our model aims to acquire.

Table 5.1: Regular grammar for TODS encoding in a language modeling problem. We have that string w in U is sampled from $u(w)$ an utterance distribution. The string w in R is sampled from a response distribution of words $r(w)$. <se*> are delimiter tokens for each segment.

$$
\begin{aligned}
S &\rightarrow UBARS \mid \epsilon \\
U &\rightarrow \texttt{<su>} w_u \sim u(w_u) \texttt{<eu>} \\
B &\rightarrow \texttt{<sb>} k \subset K_b; i \subset I_b \texttt{<eb>} \\
A &\rightarrow \texttt{<sa>} k \subset K_a; i \subset I_a \texttt{<ea>} \\
R &\rightarrow \texttt{<sr>} w_r \sim r(w_r) \texttt{<er>}
\end{aligned}
$$

Nodes U and R follow a dialog act distribution (Traum and Hinkelman, 1992; Bunt, 2006), and the literature presents many examples of corpora presenting this utterance-response structure. In the B node, we have a classification task. It is, the model should classify an intent given an utterance. A possible approach to simulate this behavior is assigning a pseudo-label for this utterance-response pair. In A we want our model to learn a policy optimization (Williams et al., 2016).

If we simulate nodes U, B, and R we have an excellent candidate grammar for an intermediate task. Table 5.2 illustrates our fabricated grammar that simulates an intermediate transfer step.

Table 5.2: Regular grammar for simulating the distribution of the target grammar. In K_c are the set of possible classes the model should classify for a given utterance. In red are removed elements, and in blue edited elements

$$
\begin{aligned}
S &\rightarrow UBARS \mid \epsilon \\
U &\rightarrow \texttt{<su>} w_u \sim u(w_u) \texttt{<eu>} \\
B &\rightarrow \texttt{<sb>} {\color{blue} k \subset K_c}; {\color{red} i \subset I_b} \texttt{<eb>} \\
A &\rightarrow \texttt{<sa>} {\color{red} k \subset K_a; i \subset I_a} \texttt{<ea>} \\
R &\rightarrow \texttt{<sr>} w_r \sim r(w_r) \texttt{<er>}
\end{aligned}
$$

We use the data obtained from online forums to simulate this simplified grammar. First, we set the message that creates the thread as an utterance and the topic presented as the classification problem. We put an empty string for action and replicate this pattern for every message replying to the original thread message as response. Finally, we append the unique tokens to encode it as shown below:

```
<sos_u>i'll be in amsterdam for a week , what would be the most
reliable sim card for data usage ?<eos_u><sos_b>amsterdam north
holland province<eos_b><sos_a> <eos_a><sos_r>take your pick .
vodafone and tmobile for starters . not all those who wander
are lost<eos_r>
```

We evaluate the introduction of this step in the curriculum and compare it with previous approaches. Below we list three different curriculum strategies for the Multi-WoZ data:

1. The first curriculum named "gpt-*/multiwoz," begins by inheriting the weights from a GPT-2 pre-trained in data set 1 from HuggingFace.

2. The second curriculum named "gpt-*/noencode/multiwoz," begins by inheriting the weights from a GPT-2 pre-trained in data set 1 from HuggingFace. We train it on TripAdvisor data, with the random ordering of the messages and no special encoding.

3. The third curriculum named "gpt-*/encode/multiwoz," begins by inheriting the weights from a GPT-2 pre-trained in data set 1 from HuggingFace. We train it on TripAdvisor data encoded as TODS, shown below.

For both curricula that include TripAdvisor, we pre-process using the same script for Multi-WoZ.

Figure 5.3 presents the curriculum and how we encode each instance. We shuffle all messages and merge them to replicate the regular recursion. In the last phase, we train on MultiWoZ without any data augmentation.

We train this architecture in the same strategy as Yang et al. (2020), where all the dialog session is presented to the model in a single sequence. The training sequence should contain at most 512 tokens, sessions with more than this maximum length we split into the following sequence. For the inference stage, we decode tokens until we reach an end-of-response token. We train on the pseudo-tods with the hyper-parameters described in Table 5.3. For all other hyper-parameters, we keep the default configuration from HF Trainer. We optimize small, medium, and large for 60k, 120k, and 240k steps, respectively, and we set these values to match our computational constraints.

Given the restrictions of GPU memory, a Quadro RTX 6000 with 24GB, we trained the small model with a batch size of 8, the medium with 4, and the large with 2. We used a gradient accumulation for all sizes, to sum up to 32 examples for the update. The training time for TripAdvisor data was 1 week, 2 weeks, and 1 month for each size, and Multiwoz was 9 hours, 2 days, and 1 week.

Parameter	Value
Learning Rate	2e-5
Weight Decay	0.1
Token Length	512
Adam β_1	0.9
Adam β_2	0.999
Adam ϵ	1e-8

Table 5.3: Hyper-parameters for training GPT-2.

We observe significant improvements from this pre-training approach. Finally, with the same configurations described in Table 5.3, we train the model on MultiWoZ data until convergence, using early stopping with the tolerance of five steps. The evaluations are performed without belief state or database interaction. We only generate responses based on belief state and system action.

5.5 Metrics

We evaluate the proposed curriculum under two perspectives: (i) the running loss for each curriculum variation; and (ii) the standard metrics for the MultiWoZ data sets, computed by Nekvinda and Dušek (2021) library. The first analysis investigates how the initialization proposed by each curriculum helps in the optimization process for minimizing the empirical loss for the target grammar. The second analysis measure, given each curriculum, how this improved minimization translates into better agent performance for the TODS task. In this step, we compute using `BLUE`, `INFORM`, and `SUCCESS`.

In the running loss, we optimize for the cross-entropy loss over the next token with teacher enforcing. For each token in the sentence of size N, we compute the loss between sentence pairs x, y as described in the equation below,

$$\ell(x, y) = L = \{l_1, \ldots, l_N\}^\top, \quad l_n = -\sum_{c=1}^{C} w_c \log \frac{\exp(x_{n,c})}{\sum_{i=1}^{C} \exp(x_{n,i})} y_{n,c} \tag{5.2}$$

where C is the vocabulary size, and $x_{n,c}$ is the difference in softmax for predicted token from x_n and real y_n.

For the high-level metrics, `BLEU`, `INFORM`, and `SUCCESS`, we use the Nekvinda and Dušek (2021) implementation. The `BLEU`, a metric originally proposed for machine-translation that decomposes sentences in n-grams to measure the fluency of original and machine-generated sentences. As in Papineni et al. (2002), we compute `BLEU` by first taking the modified precision score p_n as

$$p_n(\hat{S}; S) := \frac{\sum_{i=1}^{M} \sum_{s \in G_n(\hat{y}^{(i)})} \min(C(s, \hat{y}^{(i)}), \max_{y \in S_i} C(s, y))}{\sum_{i=1}^{M} \sum_{s \in G_n(\hat{y}^{(i)})} C(s, \hat{y}^{(i)})} \tag{5.3}$$

where $C(s, y)$ is the count of n-substrings in y that appear in $\hat{y}$. And now, we compute the brevity penalty as

$$\mathrm{BP}(\hat{S}; S) := e^{-(r/c-1)^{+}} \tag{5.4}$$

where c is the candidate string length, r is the effective reference corpus length and $(r/c-1)^{+} = \max(0, r/c-1)$ is the positive part of $r/c-1$. So we have

$$\texttt{BLEU}_w(\hat{S}; S) := \mathrm{BP}(\hat{S}; S) \cdot \exp\left(\sum_{n=1}^{\infty} w_n \ln p_n(\hat{S}; S)\right). \tag{5.5}$$

The `INFORM` metric measures if the last offered entity matches the goal constraints given the delexicalized version of the response. The `SUCCESS` rate measures the proportion of entirely successful dialogues. It is, given provided constraints for the task at the end of the dialogue, every utterance, slot, and action is precisely selected in each turn. These metrics are measured in the test set and held out of training. The MultiWoz data is, by default, divided into three sets, the train set, the dev (or validation) set, and the test set. After training on the train set, we evaluate the cross-entropy for 512 tokens in the dev set, which is held out of training.

5.6 Results

Our evaluation consists of two experimental evaluations, the curriculum convergence time and optimality. And a direct comparison with the select base methodology (Yang et al., 2020). For convergence, we vary the model size and evaluate the scalability properties of the curriculum. In this phase, we do not use any data augmentation techniques. In the base method, we directly compare standard metrics with the default curriculum using the UBAR training script, which also applies the Multi-Action Data Augmentation technique. We also compare the training under low-resource settings, training on 5, 10, 20, and 50% of the data, to evaluate the generalization capabilities provided by the curriculum.

We trained the curriculum targeting the MultiWoZ as described in previous sections for all three curriculums. In Figure 5.4, we present the loss for the validation split over time. We can observe that the curriculum with the pseudo-tods task has a significantly better starting point for the MultiWoZ task and converges to a lower loss than the other curriculums. For the small architecture, it converges faster.

In Table 5.4, we present the TODS metrics computed for the test split. We can observe that the pseudo-tods curriculum performs better than other curricula. The `COMBINED` score is given by $(\texttt{INFORM} + \texttt{SUCCESS}) * 0.5 + \texttt{BLEU}$.

Table 5.4: Standard metrics for the test split.

Curriculum	Model Size	`BLEU`	`INFORM`	`SUCCESS`	`COMBINED`
No	Small	**31.3**	93.4	**90.4**	123.2
No-encode		**31.3**	93.3	**90.4**	123.2
CTL (ours)		**31.3**	**93.6**	**90.4**	**123.3**
No	Medium	31.7	93.2	90.3	123.5
No-encode		**31.8**	93.3	90.2	123.6
CTL (ours)		31.7	**93.5**	**90.5**	**123.7**
No	Large	31.5	**93.3**	89.6	123.0
No-encode		31.0	93.0	89.3	122.2
CTL (ours)		**31.6**	**93.3**	**89.9**	**123.2**

The faster adaptation and overall loss observed during training also translated to the high-level metrics, and the larger the model, the more consistent the gain. Our approach is also appealing to low-resource settings. We present the evaluation over reduced training sets in Table 5.5. We vary from 5 to 50% of the training data. We can observe that, in general, the CTL approach produces better `BLEU`, and in the case of larger models, better `INFORM`.

Table 5.5: Standard metrics for the test split in a low-resource setting.

		5%			10%			20%			50%		
Curriculum	Size	`BLEU`	`INFORM`	`SUCCESS`	`BLEU`	`INFORM`	`SUCCESS`	`BLEU`	`INFORM`	`SUCCESS`	`BLEU`	`INFORM`	`SUCCESS`
No		20.0	**86.6**	66.7	26.0	90.6	80.7	28.5	**92.8**	**88.2**	**30.7**	93.4	89.9
No-encode	S	21.7	86.2	**72.6**	25.3	**91.7**	**82.0**	28.2	92.4	84.8	30.5	**93.6**	**90.0**
CTL (ours)		**22.7**	85.3	69.1	**26.8**	91.4	81.9	**28.8**	92.3	85.7	30.6	93.3	89.5
No		22.2	82.7	59.0	27.0	92.2	**82.7**	28.8	92.4	87.2	30.6	93.1	**89.9**
No-encode	M	23.2	**87.5**	**75.0**	26.5	**92.6**	82.6	**29.0**	92.1	86.8	30.8	93.1	89.4
CTL (ours)		**24.2**	**87.5**	72.9	**27.3**	90.5	81.0	**29.0**	**92.6**	**87.8**	**30.9**	**93.4**	**89.9**
No		**24.8**	88.7	**77.7**	**27.0**	90.9	81.9	29.2	92.2	86.0	30.8	93.2	88.6
No-encode	L	23.8	87.2	76.3	26.3	92.2	**83.9**	**29.8**	92.4	**87.6**	30.7	93.3	89.6
CTL (ours)		24.7	**91.9**	76.5	**27.0**	**92.3**	83.4	29.2	**92.6**	84.7	**31.0**	**93.5**	**90.0**

5.7 Discussion

In the loss analysis during training, we observed that our proposal has a better initialization for all model sizes and converges to a lower minimum in the validation data concerning another pre-training approach. This loss reduction also translates for the high-level metrics, `BLEU`, `INFORM`, and `SUCCESS`, which in most cases, the CTL presents a better score than known alternatives.

For the low-resource setting, we observed that our `BLEU` is better or equivalent to other approaches for every configuration. Other metrics seem to oscillate. Our proposal resembles the response generation aspect with high quality. It does not encompass the NER skills required in MultiWoZ, so it could lose some generality during the fine-tuning phase.

To investigate the generalization capabilities of the curriculum, we explored an assisted zero-shot classification for the topic under discussion. Without ever being trained to classify the topic "Rio de Janeiro," we explored if our model could complete the classification only by passing `rio` as the initial token. Figure 5.5 and Figure 5.6 show the auto-regressive generated belief and response for a sentence out of the training data for medium and large models, respectively.

Although the small model deviates a lot in syntax and semantics, the large model correctly infers our expected output. This example is evidence of the robust capabilities of LMs. This reveals that the curriculum guides the nature of our generating process.

While we only explored a single step in the curriculum, with encoding and classification, we can further improve for NER and parsing, which could result in higher gains for the TODS metrics.

5.8 Synthesis

In this chapter, we investigated our CTL approach through a low-level and high-level perspective. The cross-entropy presented a better initialization and arrived at better optima. The chatbot metrics improved in the final evaluation, so we observed a general gain for low- and high-level analysis through this methodology.

There are many possible configurations of the curriculum. We narrowed and better controlled our analysis by varying the classification task. This would lead to increased usage in computational costs, which would exceed our budget. To counter this limitation in analysis, we proposed to qualitatively analyze the generalization capabilities by carefully prompting our model with out-of-scope queries, which presented a strong ability to infer tokens in our grammar.

We believe this behavior is possible by the meta-learning skill in the massive intermediate data, which our curriculum hugely benefits.

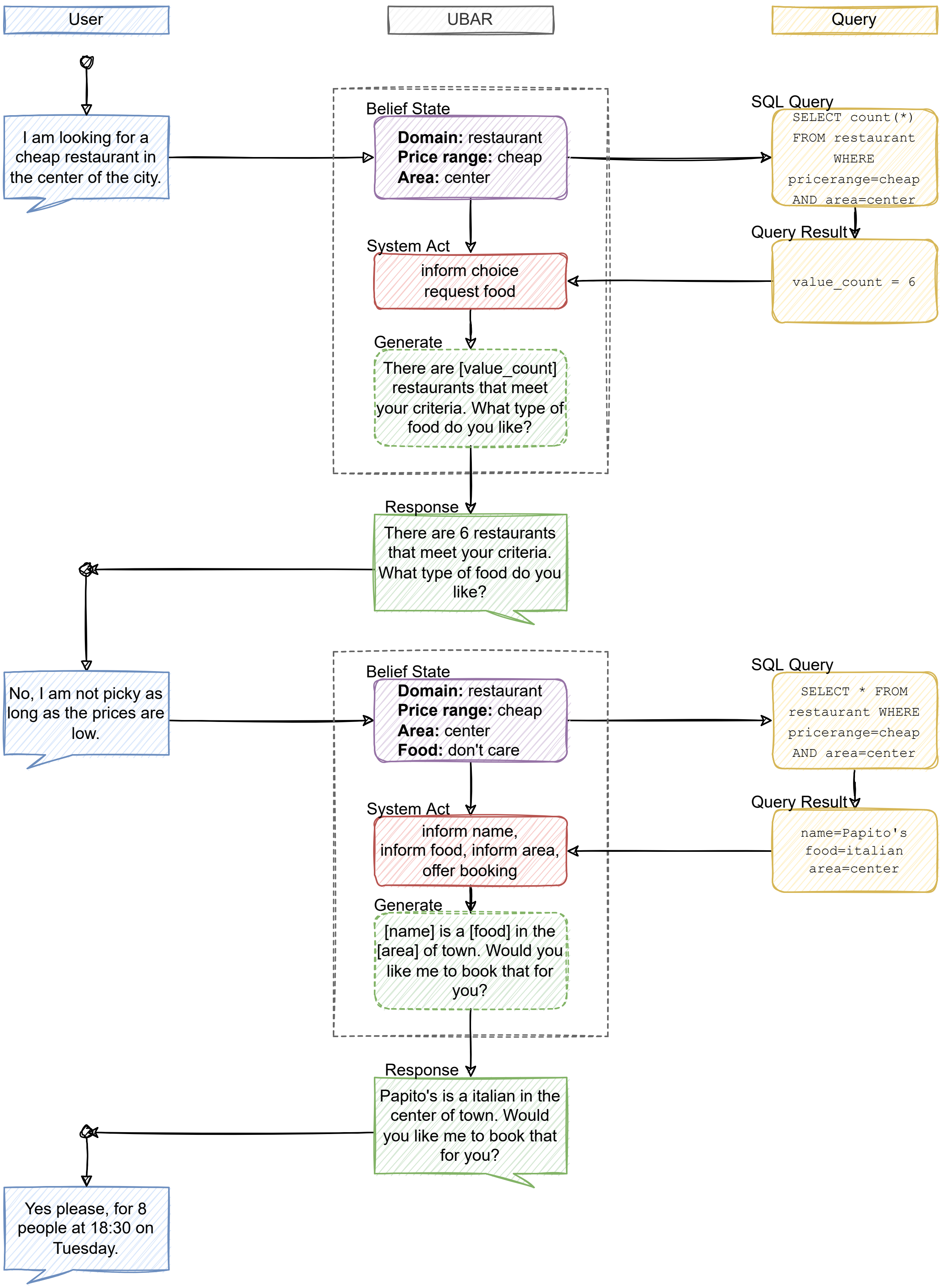


Figure 5.2: dialog interaction in the UBAR framework. The user starts a dialog, then UBAR infers intents and slot values for utterance. This belief state is then parsed for communicating with external knowledge. Then the resultant query is translated into a system action which UBAR transcribes as a response to the user.

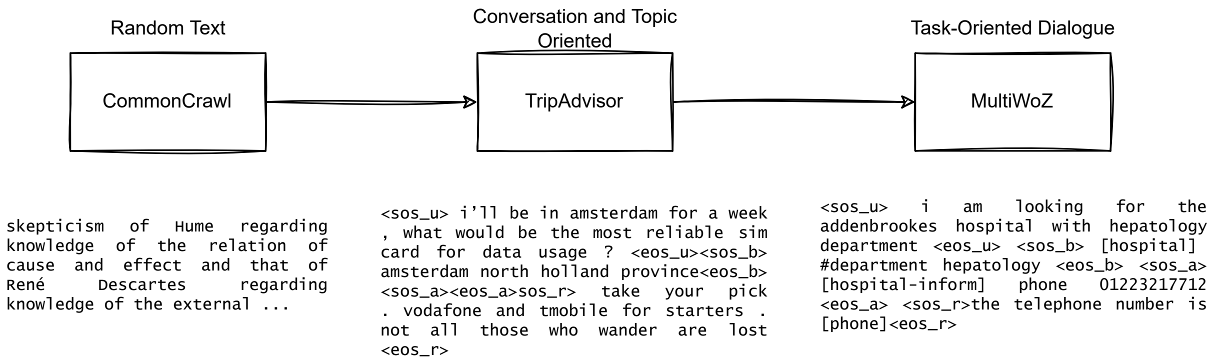


Figure 5.3: Curriculum for MultiWoZ experiments.

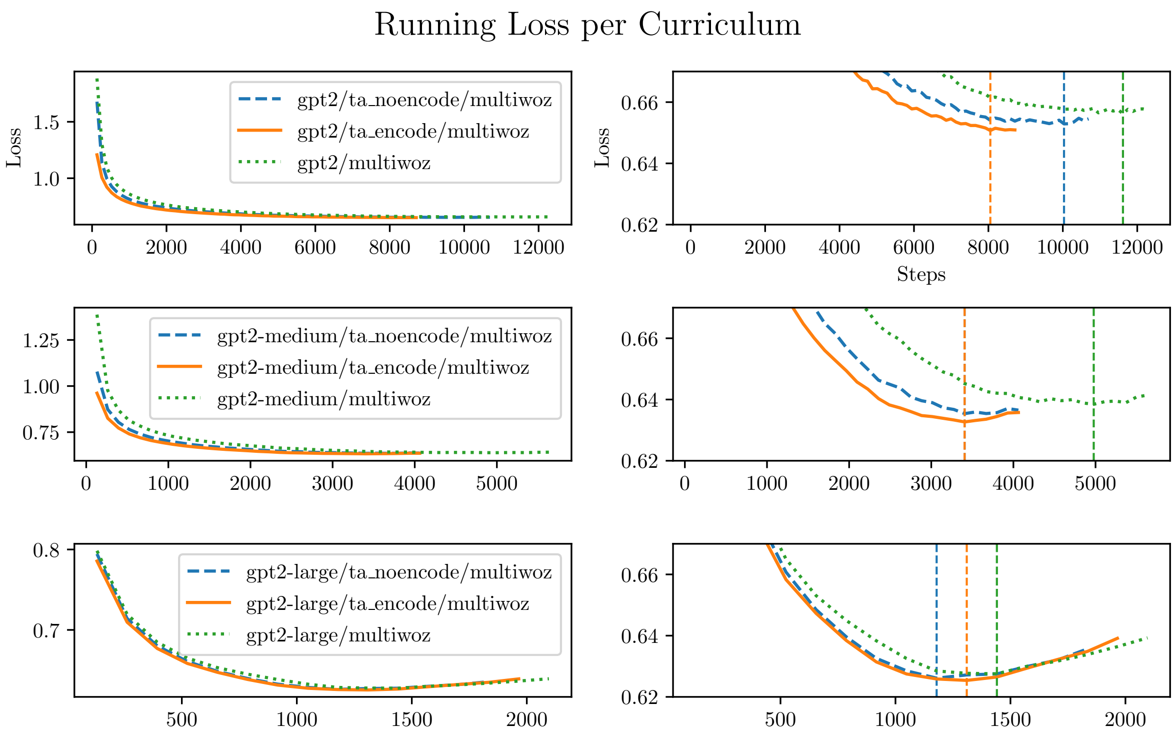


Figure 5.4: Loss in validation split for each curriculum configuration with varying model size.

Figure 5.5: Generalization capabilities of the curriculum for GPT2 (small).

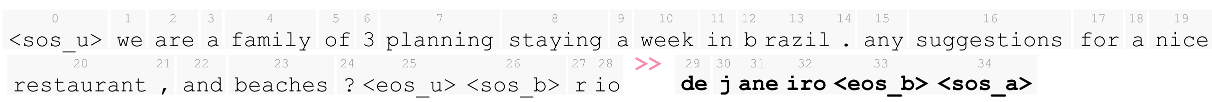


Figure 5.6: Generalization capabilities of the curriculum for GPT2 (large).

Chapter 6

Case Study

In this chapter, we present our solution on applying the CTL methodology in a simulated "real-world" scenario. We illustrate the process from the ground, starting from posing the problem, generating the data, training, and evaluating the model. This case evaluation enlighted several other aspects that should be considered when designing a real-world solution.

6.1 Context

This research was in a wider context of a project whose initiative was to research and develop new state-of-the-art tools for chatbot creation. Ci&T sponsored the project to enable the deployment of high-quality Portuguese-speaking chatbots and further improve online business in Brazil. The group comprised undergraduate and graduate students from UNICAMP. The research was carried out in many aspects for a complete chatbot creation, ranging from data generation and annotation. We studied LMs for fully end-to-end conversations.

Chatbots are an omnipresent technology in several companies in Brazil. However, they usually still use first-generation chatbots. This considers predefined menus, and the navigation is performed by the input of numerical indexes that the client wants to follow. This technology is restricted and tends not to provide a good user experience. The object of the study was to advance a more flexible and humanized chatbot solution.

The academy provides several state-of-the-art solutions for conversational agents based entirely on natural dialogs. However, there is still a gap between well-founded research datasets curated and improved by research groups and a dataset generated directly for a real-world application. Great uncertainty arises when dealing with bleeding-edge technology.

6.2 Objective

To evaluate our methodology and bridge the gap between research and implementation, we developed a real-world scenario in the Portuguese language for a complete evaluation of our methodology. We designed an experiment and implemented a fully functional chatbot for pizza ordering. The chatbot has the same objective as a human attendant in this dialog. It should guide and request the client for the ordering and requires operational information, like the delivery address and payment method. For deploying an agent with this purpose, we needed a series of solutions to create a reproducible methodology for the framework.

After concluding the dataset generation process, we evaluated our curricular methodology, which enables us to acquire a conversational agent in a low-resource setting. In this context, with only a few dozen dialogs, our solution can produce a feasible chatbot.

6.3 Protocol

Stakeholders suggested the scenario for evaluating our methodology in the project. We considered a common business in Brazil to reflect properties to investigate in a chatbot. The pizza ordering problem allows us to explore entity extraction, parsing, Natural Language Understanding (NLU), and NLG. We could explore a complete scenario with variations on pizza configuration. Nevertheless, it would only increase the data dependence without contributing to the evaluation. In this sense, we restricted topping and half separation regarding pizza. We set our scenario in the following way:

> A pizza-ordering business has three possible toppings: Pepperoni, Mozzarella, and Calabrese. When ordering, the client could choose between a pizza of one kind of topping or split, half one topping, half the other. The client could ask for any number of pizzas and combinations. The operational procedure to deliver the pizza is performed by first asking for the address, which is used to calculate the delivery fee. Then it should provide the payment method.

To generate the real-world conversational data, we developed and executed a methodology named "Teatro", which is heavily inspired by the Wizard-of-Oz approach Kelley (1984). In this framework, a human plays the role of a user, and another plays the role of an intelligent robot; the user knows that (s)he is speaking with an intelligent machine and aims to accomplish some task. By attempting to dialog with a machine, the human behaves more predictably, as it restricts the tone and the conversation model. Our methodology was composed of the following steps:

1. Schema Modeling
2. Stories Generation
3. Dialog Interaction
4. Annotation
5. Training
6. Test and Improvement

Figure 6.1 presents the overview of our method. First, we start by designing the schema in the Schema Modeling phase, which formally describes the lexical center's aspects of our conversation. A specialist in the domain lists known previous client intents when a client appeals to an attendant, the possible slot values in each intent, and how an agent should act for each kind of scenario. This initial schema, continuously updated, gives us the bootstrap to formulate possible dialog flows at a high level for Stories Generation, which is described by a Markov Decision Process (MDP).

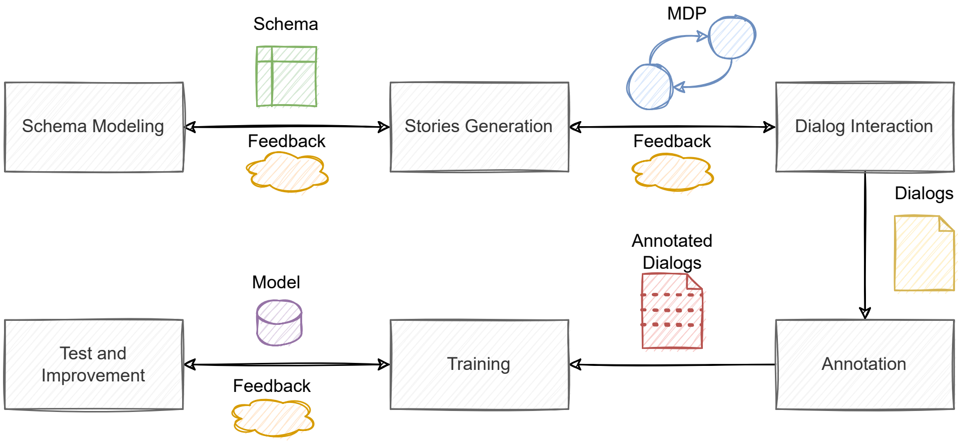


Figure 6.1: Diagram for Teatro methodology.

The MDP is now sampled to generate plots of dialogs, in which two connected agents should play each role, generating natural language dialogs. These dialogs are now annotated for missing intents and slot values for later being used in training a conversational model. To reduce the costs of annotated data, we proposed the curriculum described in Figure 6.2. The trained model is now evaluated on offline data and through interactions in a web interface, which allows us to capture misbehaviors. Finally, we deploy to clients and keep monitoring for continuous improvement.

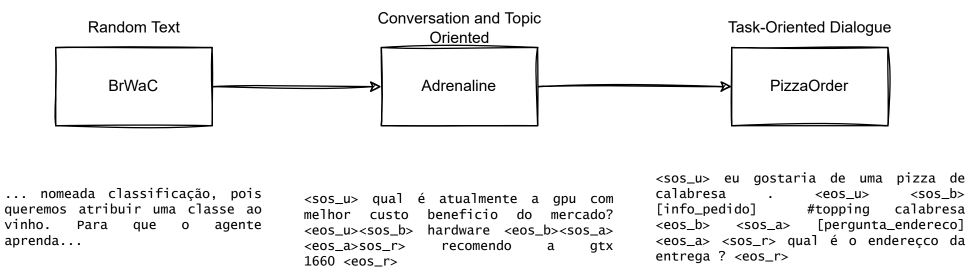


Figure 6.2: Curriculum adopted for the Case Study experiment.

In the following, we describe how each step was carried out. Note that this is not a linear flow; the steps could occur in parallel or cycle between one another to improve the modeling.

Schema Modeling

The first step into implementing a TODS in a business context is to formalize initial communication means through dialog acts. It models the conversation with clients with topics of conversation, possible user intents, keywords to track, and how we should respond to each interaction. We begin by modeling a conversation by clearly defining these concepts: intents, slot values, and actions. These entities allow us to construct possible dialog flows and recognize additional needs to be modeled to match essential business needs.

Intents are defined as user intention with an utterance. For example, when the client says "I want to order a pizza." or "I would like to buy a pizza". Although they are not written with the exact words, they share the same intent, *e.g.*, `order_pizza`. Slot values are the specification of user intent. For example, "I want a pizza of pepperoni" in this context, pepperoni is an important lexical entity we should save for generating an order, so `topping:pepperoni` is a slot value that we should save. Actions are what decisions the system takes given a policy. For example, if the client correctly described the ordering, a possible action to take is `ask_for_delivery_address`.

The set of intents, slot values, and actions a conversation could assume, we call it schema. Designing an adequate schema is fundamental in constructing conversational data because it allows us to reduce complexity and predict most aspects to implement the final TODS. A note is to keep the schema as simple as possible. In this manner, we cover enough of the conversation to correctly perform the task and get a less complex problem to learn, not to increase data dependency.

To handle basic pizza ordering, we start by constructing an initial schema following basic intuition from a client who knows what to expect from this dialog. We first enable to inform an order with an intent to "`inform_order`", a slot value for toppings "`topping:pepperoni`", and an action to get client address "`ask_delivery_address`". Then we model a MDP, described in the next section, for this schema and interactively update both MDP and schema to match an expected behavior.

By continuously interacting with the dialog MDP and the schema and reviewing real-world pizza order conversations, we updated until arriving at a flow that could handle many expected interactions. In Tables 6.1, and 6.2 below, we show the final schema.

Intents	Actions
pergunta_sabor, informa_pedido, confirmar, informa_endereco, informa_pagamento, agradecimento, fazer_pedido, confirma_pix, cumprimenta, altera_pedido, sem_troco, pergunta_endereco, informa_valor, pergunta_pagamento, confirma_pedido, informa_bandeira, via_pagamento	resposta_sabor, confirma_pedido, informa_valor, pergunta_endereco, pergunta_pagamento, informa_tempo, pergunta_pedido, informa_pix, como_ajudar, confirma_alteracao, informa_pagamento, pergunta_precisa_troco, confirma_pix, informa_pedido, cumprimenta, confirmar, pergunta_entrega, pergunta_bandeira, agradecimento, confirmar, pergunta_itens

Table 6.1: Possible intents and actions that utterances and responses for user interactions.

The resulting utterances could be grouped into four groups, general intents for handling greetings, confirmations, and usual phrases. Payment intents to handle payment information, values, etc. Operational intents to handle delivery addresses, and the last is Pizza intents to handle toppings, sizes, and models.

Slot	Description	Type	Example
type	Pizza format.	categorical	Inteira, Meia
topping	Selected pizza topping.	categorical	Calabresa, Muçarela, ...
address	The address to deliver the pizza.	textual	Rua Bragança Lote 143
payment	Payment method.	categorical	Pix, Dinheiro, ...

Table 6.2: Proposed schema model for slot values in the problem of pizza ordering.

This simplified schema has just a few slot values compared to other data sets of the exact nature. We are attempting to explore the complexity of parsing untrivial properties, like monetary values freely written in natural language, and also address where the order of elements in the address is not guaranteed.

Stories Generation

Parallel to the schema modeling iteration, we model possible dialog flows through an MDP based on past conversations. We force the user to explore many variations of the designed schema and suggest to the user many social-communication approaches to start and conduct the conversation to generate more dialogue variability. On the agent side, we orient the worker always to bring the conversation back to an expected state. This MDP is both modeled using the intuition of consumers of this kind of service and by analyzing real-world human-to-human data.

In Figure 6.3, we present the final MDP obtained in this process. Initially, we had unmerged paths that followed each kind of flow. Then on the system side, we added nodes that could merge those paths to the same cycle. In the payment path of the dialog, we attempted to ask directly for as much information as possible to simplify the resulting path.

Table 6.3 describes each node agent that should perform the interaction and the suggested utterance. We base the names and descriptions in the universal schema model on inform-request acts (Paul et al., 2019). By describing acts in an inform-request schema, we could clearly define many slots and simplify the reasoning for the general dialog flow.

After modeling the MDP, we sample stories enriched by the schema to present to the pair of workers and assign each one a role in the play. A probability is associated with each transition, so we randomly navigate the graph until reaching the final node to end the conversation. For each node with an inform-type intent, we sample the respective value from the schema shown in Table 6.2.

Dialog Interaction

The stories sampled act as a high-level screenplay, so the user has a general intuition as to what to do in the conversation but is free to express himself in the way he wants

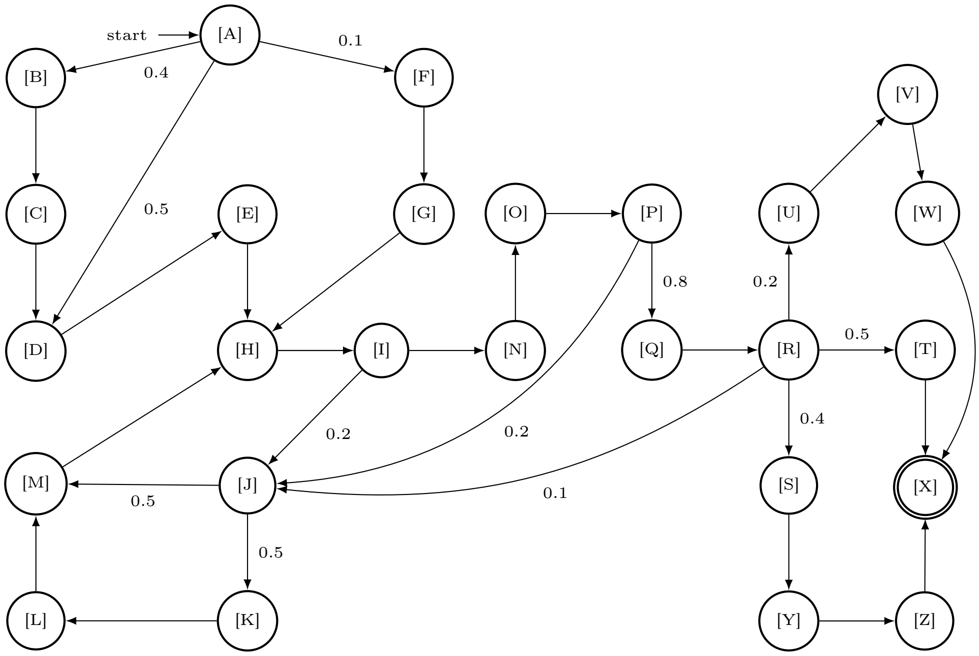


Figure 6.3: Markov Decision process for estimated dialog flows in a pizza ordering context.

to generate linguistic variability. The improvisation and interpretation aspect of this approach gives the name of the methodology, "Teatro," which is the Portuguese word for theater.

To create real-world human-to-human conversational data, we had to develop software tools to monitor and guide the conversation in a manner that covered our schema and sounded natural. This process is assisted by the Mephisto framework to handle interface, backend, and storage.

In Figure 6.4, we show our tool for data acquisition. This online interface was deployed for 20 allocated volunteers, and the system randomly assigned a screenplay and another volunteer to start the conversation. The upper-left box describes which role the volunteer should play. The lower-left window provides the "plot". It describes the information that should be exchanged in the dialog, like intents for each utterance and slot values that should be informed. After the dialog is completed, we have to annotate these dialogues.

Annotation

After the conversation ends, the framework already gives us all the slot values, as they were sampled in the previous method and most of the intents in the conversation, so a minor additional annotation for the missing intents is needed. To assist and speed up the annotation process, we also developed a web interface with a point-and-click visual interface to help workers. Figure 6.5 shows our approach for annotating dialogues. By loading the dialogues in a web interface, the user is prompted with each conversation, with clicks, and selects the user marks intents and slot values for each turn.

After the annotation process is finalized, the Web interface enables to download of a JSON that could be directly converted to training data for the learner. This is the final step to deploying the model.

Table 6.3: Nodes description for the dialog flow.

Node	Agent	Description
[A]	Control	Start
[B]	`[Client]`	Greet
[C]	`[System]`	Ask order
[D]	`[Client]`	Inform intent
[E]	`[System]`	Ask itens
[F]	`[Client]`	Ask available toppings
[G]	`[System]`	Inform toppings
[H]	`[Client]`	Inform itens
[I]	`[System]`	Confirm itens
[J]	Control	Flow for product modify
[K]	`[Client]`	Ask to modify
[L]	`[System]`	Confirm modify
[M]	`[Client]`	Inform modify
[N]	`[Client]`	Confirm
[O]	`[System]`	Inform value
[P]	`[System]`	Ask address
[Q]	`[Client]`	Inform address
[R]	`[System]`	Ask for payment method
[S]	`[Client]`	Inform type:pix
[T]	`[Client]`	Inform type:card
[U]	`[Client]`	Inform type:cash
[V]	`[System]`	Ask if needs change
[W]	`[Client]`	Inform change needed
[X]	`[System]`	Inform order status
[Y]	`[Client]`	Confirm
[Z]	`[System]`	Inform pix code

Training

Alternatively to traditional studies where English is the targeted language, we are solving a problem in the context of Brazil, so Portuguese is the targeted language. We start with general language modeling by next token prediction in Portuguese following the same approach as Generative Pre-Training (GPT-2) (Radford et al., 2019).

For the general language modeling, we used an improved version of Wagner Filho et al. (2018), in which we included many more data sources. This model is trained on these diverse corpora until convergence, which is an initialization for the next training phase.

Based on Sanches et al. (2022), we model two major Brazilian forums as pseudo-TODS. These forums encompass a wide range of topics, so their application has a general bias toward technology, entertainment, science, culture, etc. We then feed this model to the next training phase.

We train for generalized TODS on a machine-translate *MultiWoZ* dataset to Portuguese. In this phase, the LM acquire many near skills for address handling, which improves the in the final training phase.

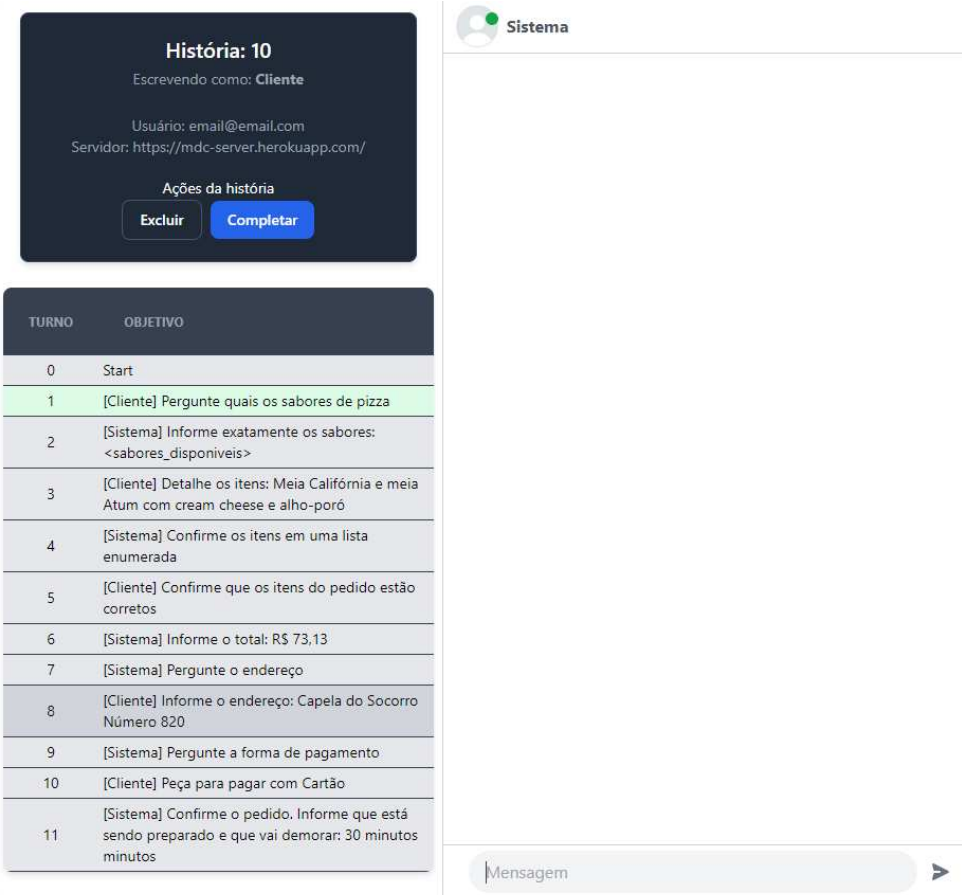


Figure 6.4: The MDC tool that implements the Teatro methodology, upper-left we have the role to be played, lower-left we have the screenplay, and in the right the message exchanging interface.

Finally, we fine-tune the model in the annotated pizza ordering data. We trained on an augmented version of the pizza ordering data for each epoch to improve address recognition capabilities. We randomly replace slot values with new values based on Zhang et al. (2020b). We use the same hyperparameters as shown in Chapter 5.

Test and Improvement

Human interaction generally is highly unpredictable and could have more variability than expected. As the conversation driven-development, we also suggest delivering the model as soon as possible for initial tests. By direct interaction with the model, we could measure the model confidence for many different types of interactions, so we could diagnose which kind of data to generate in the next batch of the Teatro. We could reanalyze our MDP to cover unexpected branching in conversation.

Some agents deviated from the expected schema during the chatting generation process, so we needed to develop different mechanisms to ensure they would not commit the same error. This was performed by a more emphatic story signaling expected behaviors, and the schema was more visually appealing.

To acquire better generalization on intent recognition and detecting slot values, we

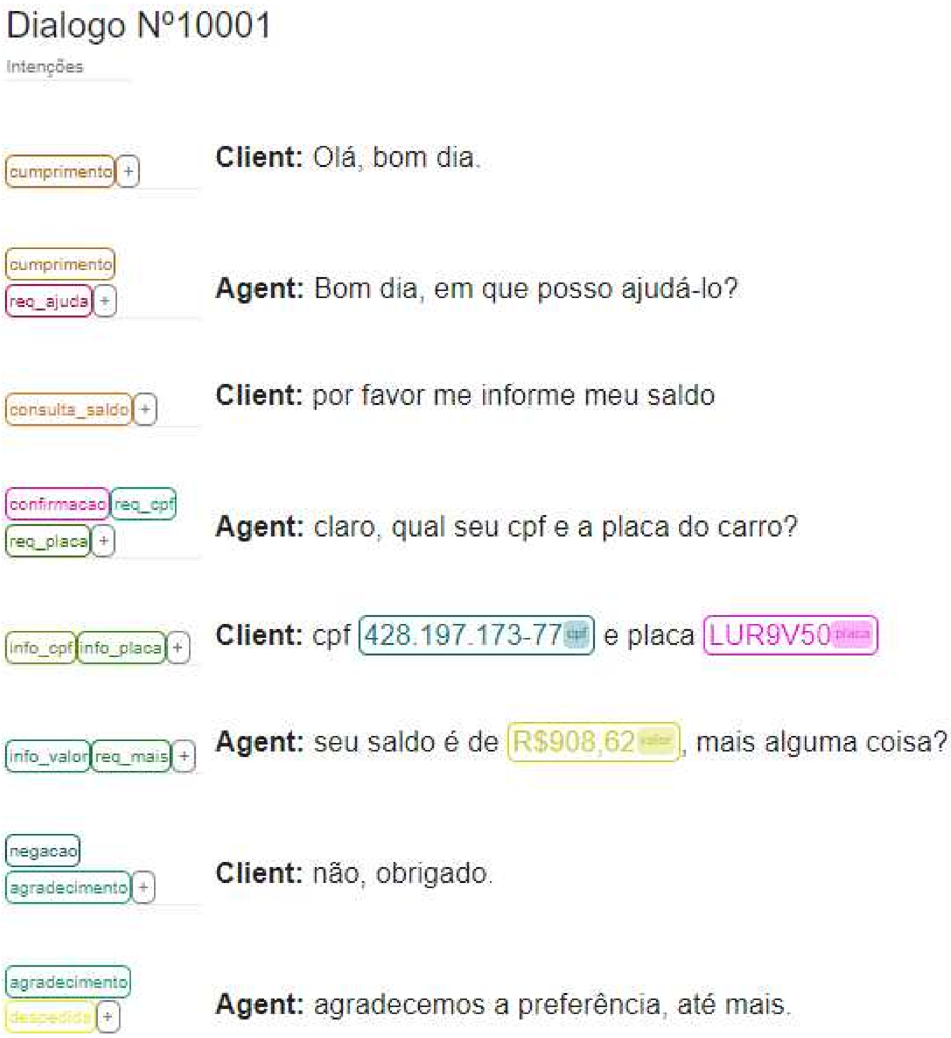


Figure 6.5: Annotation tool for TODS. With a visual interface, Assis allows one to quickly annotate dialogues without specialized knowledge.

used some techniques on data augmentation. For the intent, we explored the Zhang et al. (2020b) approach. We swapped actions that could be inferred from the same state to generate new dialog flows. For the NER augmentation, we created a natural language address generation that could be sampled for more variations of valid addresses.

Implementation

We implemented an API for ordering communication to test our solution in real-world environments. It accesses a database for prices, topping options, transactions, etc. This is suited to sending the final order as well. We implemented an interface through Telegram API so we could quickly test the agent with multiple users [1].

[1] https://github.com/ZeCariocaUnicamp/zecarioca/blob/master/telegrambot.py

6.4 Evaluation Results

To generate a reasonable amount of data, we counted the collaboration of volunteers from the project. This included direct participants from the project and other stakeholders by including Ci&T collaborators and other company Mutant (as a stakeholder from Ci&T). After a series of interactions with Teatro, we generated 70 dialogs with a mean of 15 turns between client and system, with a minimum of 9 and a maximum of 30 turns. Each utterance has an average of 8 words, totaling 8263 words for the entire set.

We first evaluate our model's ability to recognize intents and extract intents by parsing the generated response and computing the precision, recall, and F1 metrics. Table 6.4 presents the confusion matrix for the intent classification. We observe that the models correctly recognize every intent, with an adequate score in the validation data.

Table 6.4: Metrics for intent recognition.

slot	precision	recall	f1-score	support
informa_pagamento	1.0	1.0	1.0	28
sem_troco	1.0	1.0	1.0	7
informa_troco	1.0	1.0	1.0	3
fazer_pedido	1.0	1.0	1.0	46
informa_endereco	1.0	1.0	1.0	41
informa_pedido	1.0	1.0	1.0	64
cumprimenta	1.0	1.0	1.0	17
pergunta_sabor	1.0	1.0	1.0	7
confirmar	1.0	1.0	1.0	93

Table 6.5 demonstrates the model's ability to extract slot values for the test data. The model could ideally recognize addresses and monetary values given the data augmentation technique and a controlled process to generate slot values.

Table 6.5: Metrics for slot values.

slot	precision	recall	f1-score	support
itens	1.0	1.0	1.0	100
via pagamento	1.0	1.0	1.0	34
endereco	1.0	1.0	1.0	188

6.5 Discussion

We successfully developed and applied a methodology to create a pizza delivery agent from scratch based on fourth-generation dialog agents. We trained a curriculum model in a small amount of annotated examples, given its high cost to produce specialized data from scratch, as in a real-world setting.

Even with the low-resource setting, our model could always identify the desired slot values and intents in our toy data set.

We acknowledge that our generated data has a strong bias in the generation process, given the limited set of agents generating the data and the limited scope (intents and slot values). When dealing with unknown users in the data generation process, we expect much more noise and, consequently, less predictability.

6.6 Synthesis

We started this chapter by investigating the feasibility of applying the curricular transfer learning methodology to a specific context (pizza ordering) in the Portuguese language. Albeit we simplified our scope by restricting the possibilities of what kind of pizza a customer could order, we verified high quality through quantitative and qualitative testing. Our agent, in this scope, could handle many possible conversations correctly, identifying addresses and toppings.

Parallel to this, we developed a methodology that allowed a complete creation of an end-to-end TODS, following best practices and state-of-the-art tools in significant aspects of development. This allowed us to evaluate the CTL methodology in an experimental data set and provided many methodologies, tools, and empirical knowledge on chatbot creation.

Chapter 7

Conclusion

This chapter summarizes our conclusions from this research. We present our remarks, discussions, open questions, and contributions from this M.Sc. dissertation.

7.1 Overall Discussion

Differing from traditional language modeling, where the task is to predict sequences of words given a context, in a task-oriented dialog, we have a structured sequence prediction problem with three NLP sub-tasks: (i) a classification task to recognize intents; (ii) a NER to recognize entities; and (iii) a natural language generation task to predict the adequate response for a given utterance and system actions. LMs are general multi-task learners for NLP, which overfit on specific contexts with a nonstandard distribution of tokens.

In this MS.c. dissertation, we defended that having a general language modeling task as a starting point is not ideal for the problem of TODS. As our problem has a strong grammar dependency, the model should always predict the same tokens even if the dialog session has a hundred interactions.

We must construct gradual datasets to better initialize the LM and guarantee it does not degenerate in predicting the grammar for long sequences. The proposed solution, CTL, a sequence of transfer learning steps, helps to balance the trade-off between the right bias in the data and annotation costs.

We presented a theory on how this approach is not only a multi-step sequential transfer learning. Our solution can be viewed as a form of curriculum learning when considering the overall ordering within all tasks. We optimize order through the same data-generating process in the original curriculum learning approach. Our CTL approach allows us to use out-of-distribution data, which is central to learning scalability and generality for modern NLP.

Our approach only explored the CTL on a very narrow case, given that sequence-to-sequence task encoding is a relatively new proposal (Hosseini-Asl et al., 2020). However, prompt-based methods are a recent trending topic in NLP. These methods allow a pre-trained LM to perform out-of-the-box several tasks they were not explicitly trained for (also called zero-shot learning). Although those models perform pretty well in classic NLP tasks, those with a complex morphological structure still need additional learning. We could extend those models with CTL to other complex applications by creating pseudo-labeled datasets in a super-scale.

7.2 Contributions

In this research, we developed a methodology to train LMs for complex grammar acquisition. This was accomplished by training the model in intermediate datasets following the grammar simplification method. This allows external data sources to encode as an intermediate task to improve the general optima.

Unlike earlier attempts to use additional data depending on annotated data, which is expensive and not feasible for many business models, our approach used unsupervised data to structure pseudo-supervised data and model sharing to significantly reduce the costs of maintaining this kind of system. Another essential aspect of our proposal is that the data sources are not English-dependent (the default for this literature). In this sense, we could crawl unsupervised data from forums in most languages. In our case study, we successfully applied the method for the Portuguese language based on Brazilian conversational forums.

Training costs of domain-specific models are expensive, and sharing specialized pre-trained models allows many businesses to perform cheaper fine-tuning. We can fine-tune a general pizza ordering model for specific address handling in a small city in Brazil, so many small businesses could use the same pre-trained model. We summarize our direct scientific contribution as follows:

- The CTL framework for gradual grammar adaptation. We explored its grammar acquisition capabilities and experimentally evaluated them for two different contexts.

Additionally, to make this research possible, we developed several auxiliary methodologies and software tools:

- We crawled and cleaned the *TripAdvisor* data, publicly providing the processed data and crawling scripts. This was relevant for evaluating our proposal against *MultiWoZ*.

- A pizza ordering dataset in Portuguese is freely available for extending and future research.

- A methodology, named “Teatro”, to generate human-to-human conversational data based on best pratices from literature.

- To evaluate our methodology in the Portuguese context, we contributed with a web interface that implements the Teatro methodology. The software tool was built upon the Mephisto framework.

- Produced unsupervised, conversational-based data extracted from Adrenaline and Outerspace for evaluation in Portuguese.

- We created "Assis", a web-interface to quickly annotate conversational data for TODS.
- A paraphrasing methodology based on a linguistic reward used in the Teatro methodology.
- Several pre-trained models in English and Portuguese for general and conversational applications.

All methodology, tools, and models are available at `https://www.notion.so/unicamp-cit/CI-T-Unicamp-ca7a3fc10d9a43f09a14c5dd4d31e554`

7.3 Future Studies

Recent studies Delétang et al. (2022) demonstrated that Transformers are inefficient for learning regular grammar. Memory-augmented networks could almost perfectly learn any regular grammar. A possible research path is to evaluate modern augmented memory networks trained in this curriculum if it better generalizes for unknown cases, *e.g.*, and zero-shot learning.

The CTL approach enriches models' capabilities and generalization and exposes the model to out-of-scope domains inside the same meta-structure. This may lead to failure in more diverse ways. Further investigations are necessary to understand how to reduce variability and control TODS based on LMs given that they do not "clearly fail" as pipeline approaches.

Large LM are known for their capabilities of meta-learning complex tasks through unsupervised training. The CTL approach, by design, meta-teaches desired properties for a TODS agent. Our empirical results demonstrate that the larger the model, the better it learns the pseudo-tasks in the curriculum. In this sense, a Large LM could generalize, allowing few-shot, a fundamental property for agents in low-setting environments.

We applied our curriculum approach specifically for the task of TODS. We believe it is extendable for any sequence-to-sequence task based on complex grammar. The field of sequence-to-sequence modeling is relatively new. To our knowledge, no other approach uses particular encoding grammar. Still, we expect to extend our study to other domains and tasks.

Bibliography

Daron Acemoglu and Pascual Restrepo. Robots and jobs: Evidence from us labor markets. *Journal of Political Economy*, 128(6):2188–2244, 2020.

Eleni Adamopoulou and Lefteris Moussiades. Chatbots: History, technology, and applications. *Machine Learning with Applications*, 2:100006, 2020.

Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. Towards a human-like open-domain chatbot, 2020.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2015.

Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. Explaining neural scaling laws. *ArXiv*, abs/2102.06701, 2021.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

Irena Bojanova. The digital revolution: What's on the horizon? *IT Professional*, 16:8–12, 2014.

Antoine Bordes, Y. Lan Boureau, and Jason Weston. Learning end-to-end goal-oriented dialog. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, pages 1–15, 2017.

S Bozinovski and A Fulgosi. The influence of pattern similarity and transfer learning upon training of a base perceptron b2. In *Proceedings of Symposium Informatica*, pages 3–121, 1976.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, I. Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. Multiwoz - a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *EMNLP*, 2018.

Harry Bunt. Dimensions in dialogue act annotation. In *LREC*, 2006.

B. Byrne, K. Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Daniel Duckworth, Semih Yavuz, Ben Goodrich, Amit Dubey, A. Cedilnik, and Kyu-Young Kim. Taskmaster-1: Toward a realistic and diverse dialog dataset. In *EMNLP/IJCNLP*, 2019.

Kyunghyun Cho, B. V. Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. *ArXiv*, abs/1406.1078, 2014.

Radostin Cholakov and Todor Kolev. Efficient task-oriented dialogue systems with response selection as an auxiliary task. *arXiv preprint arXiv:2208.07097*, 2022.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *ArXiv*, abs/2110.14168, 2021.

Ryan Cotterell, Sabrina J. Mielke, Jason Eisner, and Brian Roark. Are all languages equally hard to language-model? In *NAACL-HLT*, 2018.

Antonella Cugini, Antonella Carù, and Fabrizio Zerbini. The cost of customer satisfaction: a framework for strategic cost management in service industries. *European Accounting Review*, 16(3):499–530, 2007.

Yinpei Dai, Hangyu Li, Yongbin Li, Jian Sun, Fei Huang, Luo Si, and Xiaodan Zhu. Preview, attend and review: Schema-aware curriculum learning for multi-domain dialog state tracking. *arXiv preprint arXiv:2106.00291*, 2021.

Donald Dansereau. The development of a learning strategies curriculum. *Learning strategies*, pages 1–29, 1978.

Norman F Davies. Language acquisition, language learning and the school curriculum. *System*, 8(2):97–102, 1980.

Grégoire Delétang, Anian Ruoss, Jordi Grau-Moya, Tim Genewein, Li Kevin Wenliang, Elliot Catt, Marcus Hutter, Shane Legg, and Pedro A Ortega. Neural networks and the chomsky hierarchy. *arXiv preprint arXiv:2207.02098*, 2022.

J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

Jacob Eisenstein. *Introduction to natural language processing*. MIT press, 2019.

Jeffrey L Elman. Learning and development in neural networks: The importance of starting small. *Cognition*, 48(1):71–99, 1993.

Meng Fang, Tianyi Zhou, Yali Du, Lei Han, and Zhengyou Zhang. Curriculum-guided hindsight experience replay. *Advances in neural information processing systems*, 32, 2019.

Francesco Foglino and Matteo Leonetti. An optimization framework for task sequencing in curriculum learning. *2019 Joint IEEE 9th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, pages 207–214, 2019.

Mary Ellen Foster. Natural language generation for social robotics: opportunities and challenges. *Philosophical Transactions of the Royal Society B*, 374(1771):20180027, 2019.

Jianfeng Gao, Michel Galley, and Lihong Li. Neural approaches to conversational ai. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference Tutorial Abstracts*, pages 2–7, 2018. doi: 10.18653/v1/p18-5002.

Maria Teresa Guasti. *Language acquisition: The growth of grammar*. MIT press, 2017.

Wanwei He, Yinpei Dai, Yinhe Zheng, Yuchuan Wu, Zheng Cao, Dermot Liu, Peng Jiang, Min Yang, Fei Huang, Luo Si, et al. Galaxy: A generative pre-trained model for task-oriented dialog with semi-supervised learning and explicit policy injection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10749–10757, 2022.

Dirk Helbing. Societal, economic, ethical and legal challenges of the digital revolution: from big data to deep learning, artificial intelligence, and manipulative technologies. In *Towards digital enlightenment*, pages 47–72. Springer, 2019.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Xiaodong Song. Pretrained transformers improve out-of-distribution robustness. In *ACL*, 2020.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Xiaodong Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *ArXiv*, abs/2009.03300, 2021a.

Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. Cuad: An expert-annotated nlp dataset for legal contract review. *ArXiv*, abs/2103.06268, 2021b.

T. J. Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B. Brown, Prafulla Dhariwal, Scott Gray, Chris Hallacy, Benjamin Mann, Alec Radford, Aditya Ramesh, Nick Ryder, Daniel M. Ziegler, John Schulman, Dario Amodei, and Sam McCandlish. Scaling laws for autoregressive generative modeling. *ArXiv*, abs/2010.14701, 2020.

Danny Hernandez, Jared Kaplan, T. J. Henighan, and Sam McCandlish. Scaling laws for transfer. *ArXiv*, abs/2102.01293, 2021.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. A simple language model for task-oriented dialogue. *arXiv preprint arXiv:2005.00796*, 2020.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *ACL*, 2018.

Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. Challenges in building intelligent open-domain dialog systems. *ArXiv*, may 2019. URL `https://arxiv.org/abs/1905.05709`.

Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. Challenges in building intelligent open-domain dialog systems. *ACM Transactions on Information Systems (TOIS)*, 38(3): 1–32, 2020.

Marcus Hutter. Learning curve theory. *ArXiv*, abs/2102.04074, 2021.

Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. *ArXiv*, abs/1406.2227, 2014.

Hyunmin Jeon and Gary Geunbae Lee. Domain State Tracking for a Simplified Dialogue System. *ArXiv*, mar 2021.

Dan Jurafsky and James H. Martin. *Speech and Language Processing, 3rd Edition.* Prentice Hall, 2020.

Jared Kaplan, Sam McCandlish, T. J. Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeff Wu, and Dario Amodei. Scaling laws for neural language models. *ArXiv*, abs/2001.08361, 2020.

John F Kelley. An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Information Systems (TOIS)*, 2(1): 26–41, 1984.

Boeun Kim, Dohaeng Lee, Sihyung Kim, Yejin Lee, Jin-Xia Huang, Oh-Woog Kwon, and Harksoo Kim. Document-grounded goal-oriented dialogue systems on pre-trained language model with diverse input representation. *Proceedings of the 1st Workshop on Document-grounded Dialogue and Conversational Question Answering (DialDoc 2021)*, 2021.

Natalia L Komarova, Partha Niyogi, and Martin A Nowak. The evolutionary dynamics of grammar acquisition. *Journal of theoretical biology*, 209(1):43–59, 2001.

Jonáš Kulhánek, Vojtěch Hudeček, Tomáš Nekvinda, and Ondřej Dušek. Augpt: Auxiliary tasks and data augmentation for end-to-end dialogue with pre-trained language models. *ArXiv*, feb 2021. URL `http://arxiv.org/abs/2102.05126`.

Hwaran Lee, Seokhwan Jo, HyungJun Kim, Sangkeun Jung, and Tae-Yoon Kim. Sumbt+larl: End-to-end neural task-oriented dialog system with reinforcement learning. *ArXiv*, abs/2009.10447, 2020.

Yohan Lee. Improving end-to-end task-oriented dialog system with a simple auxiliary task. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1296–1303, 2021.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models, 2022. URL `https://arxiv.org/abs/2206.14858`.

Xiujun Li, Zachary C Lipton, Bhuwan Dhingra, Lihong Li, Jianfeng Gao, and Yun-Nung Chen. A user simulator for task-completion dialogues. *arXiv preprint arXiv:1612.05688*, 2016.

Zhaojiang Lin, Andrea Madotto, Genta Indra Winata, and Pascale Fung. Mintl: Minimalist transfer learning for task-oriented dialogue systems. In *EMNLP*, 2020.

Bing Liu and Ian Lane. End-to-end learning of task-oriented dialogs. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 67–73, New Orleans, Louisiana, USA, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-4010. URL `https://www.aclweb.org/anthology/N18-4010`.

Jinglin Liu, Yi Ren, Xu Tan, Chen Zhang, Tao Qin, Zhou Zhao, and Tie-Yan Liu. Task-level curriculum learning for non-autoregressive neural machine translation. *ArXiv*, abs/2007.08772, 2020.

Qi Liu, Lei Yu, Laura Rimell, and Phil Blunsom. Pretraining the Noisy Channel Model for Task-Oriented Dialogue. *ArXiv*, mar 2021a. URL `http://arxiv.org/abs/2103.10518`.

Sihong Liu, Jinchao Zhang, Keqing He, Weiran Xu, and Jie Zhou. Scheduled dialog policy learning: An automatic curriculum learning framework for task-oriented dialog system. In *FINDINGS*, 2021b.

Yi Liu and Shuiwang Ji. A multi-stage attentive transfer learning framework for improving covid-19 diagnosis. *arXiv preprint arXiv:2101.05410*, 2021.

Nurul Lubis, Christian Geishauser, Michael Heck, Hsien-chin Lin, Marco Moresi, Carel van Niekerk, and Milica Gasic. LAVA: Latent Action Spaces via Variational Autoencoding for Dialogue Policy Optimization. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 465–479, Stroudsburg, PA, USA, nov 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.41.

Richard Luscombe. Guardian google engineer put on leave after saying ai chatbot has become sentient. `https://www.theguardian.com/technology/2022/jun/12/google-engineer-ai-bot-sentient-blake-lemoine`, 2022. Accessed: 2022-09-05.

Andrea Madotto, Chien Sheng Wu, and Pascale Fung. MEM2Seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 1:1468–1478, 2018. doi: 10.18653/v1/p18-1136.

Binyamin Manela and Armin Biess. Curriculum learning with hindsight experience replay for sequential object manipulation tasks. *Neural Networks*, 145:260–270, 2022.

Sebastian J. Mielke, Ryan Cotterell, Kyle Gorman, Brian Roark, and Jason Eisner. What kind of language is hard to language-model? *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019. doi: 10.18653/v1/p19-1491. URL `http://dx.doi.org/10.18653/v1/P19-1491`.

Nikola Mrkšić, Diarmuid O Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. Neural belief tracker: Data-driven dialogue state tracking. *arXiv preprint arXiv:1606.03777*, 2016.

Sanmit Narvekar, Jivko Sinapov, Matteo Leonetti, and Peter Stone. Source task creation for curriculum learning. In *Proceedings of the 2016 international conference on autonomous agents & multiagent systems*, pages 566–574, 2016.

Ljubica Nedelkoska and Glenda Quintini. Automation, skills use and training. *OECD Social, Employment and Migration Working Papers*, 2018. doi: https://doi.org/https://doi.org/10.1787/2e2f4eea-en. URL `https://www.oecd-ilibrary.org/content/paper/2e2f4eea-en`.

Tomáš Nekvinda and Ondřej Dušek. Shades of bleu, flavours of success: The case of multiwoz. *arXiv preprint arXiv:2106.05555*, 2021.

David Nunan. Communicative tasks and the language curriculum. *TESOL quarterly*, 25 (2):279–295, 1991.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

Shachi Paul, Rahul Goel, and Dilek Hakkani-Tür. Towards universal dialogue act tagging for task-oriented dialogues. *arXiv preprint arXiv:1907.03020*, 2019.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. SOLOIST: Building Task Bots at Scale with Transfer Learning and Machine Teaching. *Transactions of the Association for Computational Linguistics*, 9:807–824, aug 2020. ISSN 2331-8422. doi: 10.1162/tacl_a_00399.

Anastasia Pentina, Viktoriia Sharmanska, and Christoph H Lampert. Curriculum learning of multiple tasks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5492–5500, 2015.

Alec Radford, Rafal Józefowicz, and Ilya Sutskever. Learning to generate reviews and discovering sentiment. *ArXiv*, abs/1704.01444, 2017.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *ArXiv*, abs/1910.10683, 2020.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M Smith, et al. Recipes for building an open-domain chatbot. *arXiv preprint arXiv:2004.13637*, 2020.

Sebastian Ruder. *Neural transfer learning for natural language processing*. PhD thesis, NUI Galway, 2019.

Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2010.

Atsushi Saito. Curriculum learning based on reward sparseness for deep reinforcement learning of task completion dialogue management. In *Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI*, pages 46–51, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5707. URL https://aclanthology.org/W18-5707.

Matheus Ferraroni Sanches, Jáder MC de Sá, Allan Mariano de Souza, Diego A Silva, Rafael R de Souza, Júlio Cesar dos Reis, and Leandro A Villas. Mccd: Generating human natural language conversational datasets. In *ICEIS (2)*, pages 247–255, 2022.

Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. Agenda-based user simulation for bootstrapping a pomdp dialogue system. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 149–152, 2007.

Sebastian Schuster, Rushin Shah, Sonal Gupta, and Mike Lewis. Cross-lingual transfer learning for multilingual task oriented dialog. In *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, volume 1, pages 3795–3805, oct 2019. ISBN 9781950737130. doi: 10.18653/v1/n19-1380. URL http://arxiv.org/abs/1810.13327.

Weiyan Shi, Kun Qian, Xuewei Wang, and Zhou Yu. How to build user simulators to train rl-based dialog systems. *arXiv preprint arXiv:1909.01388*, 2019.

Burrhus F Skinner. Reinforcement today. *American Psychologist*, 13(3):94, 1958.

Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. Curriculum self-paced learning for cross-domain object detection. *Computer Vision and Image Understanding*, 204:103166, 2021.

Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. Curriculum learning: A survey. *International Journal of Computer Vision*, pages 1–40, 2022.

Yixuan Su, Lei Shu, Elman Mansimov, Arshit Gupta, Deng Cai, Yi-An Lai, and Yi Zhang. Multi-Task Pre-Training for Plug-and-Play Task-Oriented Dialogue System. *ArXiv*, sep 2021. URL `http://arxiv.org/abs/2109.14739`.

Haipeng Sun, Junwei Bao, Youzheng Wu, and Xiaodong He. Bort: Back and denoising reconstruction for end-to-end task-oriented dialog, 2022. URL `https://arxiv.org/abs/2205.02471`.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long range arena: A benchmark for efficient transformers. *ArXiv*, abs/2011.04006, 2021.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.

David R Traum and Elizabeth A Hinkelman. Conversation acts in task-oriented spoken dialogue. *Computational intelligence*, 8(3):575–599, 1992.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *ArXiv*, abs/1706.03762, 2017.

Jorge A Wagner Filho, Rodrigo Wilkens, Marco Idiart, and Aline Villavicencio. The brwac corpus: a new open resource for brazilian portuguese. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018)*, 2018.

Xin Wang, Yudong Chen, and Wenwu Zhu. A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni. Generalizing from a few examples. *ACM Computing Surveys (CSUR)*, 53:1 – 34, 2020.

Tsung Hsien Wen, David Vandyke, Nikola Mrkšíc, Milica Gašíc, Lina M. Rojas-Barahona, Pei Hao Su, Stefan Ultes, and Steve Young. A network-based end-to-end trainable task-oriented dialogue system. In *15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017 - Proceedings of Conference*, volume 1, pages 438–449. Association for Computational Linguistics (ACL), apr 2017. ISBN 9781510838604. doi: 10.18653/v1/e17-1042. URL `http://arxiv.org/abs/1604.04562`.

L. White. *Second Language Acquisition and Universal Grammar*. Cambridge Textbooks in Linguistics. Cambridge University Press, 2003. ISBN 9780521796477. URL `https://books.google.com.br/books?id=2Q25nuAQ-oQC`.

Jason D Williams, Antoine Raux, and Matthew Henderson. The dialog state tracking challenge series: A review. *Dialogue & Discourse*, 7(3):4–33, 2016.

Chien-Sheng Wu, Steven C.H. Hoi, Richard Socher, and Caiming Xiong. TOD-BERT: Pre-trained natural language understanding for task-oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 917–929, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.66. URL `https://aclanthology.org/2020.emnlp-main.66`.

Yunyi Yang, Yunhao Li, and Xiaojun Quan. Ubar: Towards fully end-to-end task-oriented dialog systems with gpt-2. *arXiv preprint arXiv:2012.03539*, 2020.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *ACL*, 2019.

Jian-Guo Zhang, Kazuma Hashimoto, Chien-Sheng Wu, Yao Wan, Philip S Yu, Richard Socher, and Caiming Xiong. Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking. *arXiv preprint arXiv:1910.03544*, 2019.

Yichi Zhang, Zhijian Ou, Min Hu, and Junlan Feng. A Probabilistic End-To-End Task-Oriented Dialog Model with Latent Belief States towards Semi-Supervised Learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9207–9219. Association for Computational Linguistics, sep 2020a. doi: 10.18653/v1/2020.emnlp-main.740.

Yichi Zhang, Zhijian Ou, and Zhou Yu. Task-oriented dialog systems that consider multiple appropriate responses under the same context. In *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, pages 9604–9611, nov 2020b. ISBN 9781577358350. doi: 10.1609/aaai.v34i05.6507. URL `https://arxiv.org/abs/1911.10484`.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and W. Dolan. Dialogpt : Large-scale generative pre-training for conversational response generation. In *ACL*, 2020c.

Zheng Zhang, Ryuichi Takanobu, Qi Zhu, Minlie Huang, and Xiaoyan Zhu. Recent advances and challenges in task-oriented dialog systems. *Science China Technological Sciences*, pages 1–17, 2020d.

Yang Zhao, Hua Qin, Wang Zhenyu, Changxi Zhu, and Shihan Wang. A versatile adaptive curriculum learning framework for task-oriented dialogue policy learning. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 711–723, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-naacl.54. URL `https://aclanthology.org/2022.findings-naacl.54`.

Hui Zhu, Yangyang Zhao, and Hua Qin. Cold-started curriculum learning for task-oriented dialogue policy. *2021 IEEE International Conference on e-Business Engineering (ICEBE)*, pages 100–105, 2021a.

Qingqing Zhu, Xiuying Chen, Pengfei Wu, Junfei Liu, and Dongyan Zhao. Combining curriculum learning and knowledge distillation for dialogue generation. In *EMNLP*, 2021b.