An Exploration of Dialog Act Classification in Open-domain Conversational Agents and the Applicability of Text Data Augmentation

by

Maliha Sultana

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science
University of Alberta

Abstract

Recognizing dialog acts of users is an essential component in building successful conversational agents. In this work, we propose a dialog act (DA) classifier for two of our open domain conversational agents. For this, we curated a high-quality, multi-domain dataset with ∼24k user utterances labelled into 8 suitable DAs. Our fine-tuned BERT-based model outperforms the baseline SVM classifier by achieving state-of-the-art accuracy on the proposed dataset. Moreover, it generalizes well on unseen data. To address the issue of data scarcity when training DA classifiers, we implemented different data augmentation techniques and compared their performance. Our extensive experiments show that, in a simulated low data regime with only 10 examples per label, methods as simple as synonym replacement can double the size of the existing training data and boost accuracy of our DA classifier by ∼8%. Lastly, we demonstrate how our proposed classifier and augmentation techniques can be adapted to effectively detect dialog acts in languages other than English.

Preface

Parts of Chapter 2 and 4 related to the creation of our dataset with 8 dialog acts and the development of our DA classifier has been submitted to The 25th International Conference on Big Data Analytics and Knowledge Discovery (DAWAK 2023) as a long paper titled ‘Exploring Dialog Act Recognition in Open Domain Conversational Agents’.

“Courage doesn’t always roar. Sometimes it is the little voice at the end of the day that says ‘I’ll try again tomorrow’”

-Mary Anne Radmacher

To my mom, Saima Safrin,

whose endless support and sacrifices have brought me to where I am today

Acknowledgements

First and foremost, I would like to thank my supervisor, Professor Osmar R. Zaíane for his constant support and guidance throughout the whole process. Thank you for trusting me with the project and for helping me expand it further. Moreover, I am thankful to the wonderful team of MIRA and ANA. Our weekly meetings were always insightful and something I eagerly looked forward to.

During my two years here in Edmonton, I was fortunate enough to make some amazing friends. I would whole-heartedly like to thank you guys for making my grad school experience so much fun. I will forever cherish these memories!

I would also like to thank the professors, Carrie Demmans Epp and Maria Cutumisu, who were part of the committee and who took the time to read my manuscript and assess my work.

Lastly, I am thankful to my little brother, Tanvir and my cute dog, Poko for their constant encouragement.

Table of Contents

List of Tables

List of Figures

Abbreviations

AI Artificial Intelligence.

BT Back-translation.

DA Data Augmentation.

DAC Dialog Act.

EDA Easy Data Augmentation.

LR Logistic Regression.

MT Machine Translation.

NL Natural Language.

NLI Natural Language Inference.

NLP Natural Language Processing.

NMT Neural Machine Translation.

POS Parts of Speech.

QA Question Answering.

RD Random Deletion.

RI Random Insertion.

RS Random Swap.

seq2seq Sequence to sequence.

SOTA state-of-the-art.

SR Synonym Replacement.

Chapter 1

Introduction

Human beings are inherently social. We like to exchange our thoughts and ideas through communication. Without it, the evolution of government, art, clothing and much more would not have been possible [17]. It is the building block of our relationships. Through frequent conversations, we convey our intentions, thoughts and opinions to our peers and get things done [18]. Thus, as humans, we grow accustomed to the everyday sentences we utter and the dialog acts we perform (assert, inquire, order, etc.). In linguistics and particularly in natural language understanding, a dialog act is an utterance, in the context of a conversational dialog, which serves a function in the dialog [19]. Types of dialog acts include a question, a statement, or a request for action. Dialog acts are a type of speech act. In the philosophy of language and linguistics, speech act is something expressed by an individual that simultaneously presents an information and performs an action [20]. For example, the phrase 'I would like some sugar; could you please pass it to me?' is considered a speech act as it expresses the speaker's desire to acquire the sugar, as well as presenting a request that someone passes the sugar to them. Effective communication often relies on recognizing the different dialog acts conveyed by each utterance and responding accordingly. For example, someone asking a question expects an answer as a response whereas someone giving an order expects its execution or an acknowledgment. However, it is not a trivial task given that the form and content of an utterance frequently depends on a number of factors. Two utterances might express different intents based on how the speaker delivers them and on what context. For example: ‘He wants a job.’ and ‘He wants a job?’ expect very different actions from the interlocutor (first one is stated as a fact and second one demands a yes/no answer). Moreover, sometimes it becomes relatively difficult to categorize a certain utterance. For example: ‘Let’s go for a walk’ might be a suggestion or an order. Although dialog acts are essentials of communicative life, they only became a topic of interest in the middle of the 20^{th} century not only within Philosophy, but also in other scholarly disciplines like Artificial Intelligence (AI) [21].

The attempt to mimic human conversations using AI dates back to 1966 with the advent of Eliza, a chatbot that uses pattern matching and substitution methodology to simulate conversation [22] (see Appendix A for an overview of the history and types of conversational agents). Although intended to be a mere caricature of human conversation, users were soon treating ELIZA like a companion- confiding their most intimate thoughts. Nowadays, with the development in machine learning algorithms, such as deep learning and neural networks, AI tasks like Natural Language Generation, Speech Recognition, Text to Speech Synthesis, and Sentiment Analysis have gained momentum. At present, chatbots are used as virtual assistants in different fields to enhance productivity and reduce service costs. Recent studies have found that, users often consider chatbots as friendly companions and not just mere assistants. In fact, over 40% of user requests have been observed to be emotional rather than informative [23]. How much trust a chatbot gains from its users depends on how human-like the chatbot is, that is, how efficiently and effectively it can handle natural language. As a result, recognizing the dialog act of users to generate appropriate responses has become an integral component in building a successful conversational agent. As commercial spoken dialog systems became a reality in 1999, research on classifying dialog acts have increased. A dialog system usually includes a taxonomy of dialog types or tags that classify the different functions dialog acts can play.

The Switchboard Dialog Act Corpus (SwDA) [24] is popularly used for training such dialog systems in open-domain setting. It is a large hand-labeled database of 1155 two-sided telephone conversations with provided topics. It has a total of 43 dialog acts in its taxonomy (e.g, Statement-non-opinion, Statement-opinion, Appreciation, Yes-No-Question, Wh-Question). The tags summarize syntactic, semantic, and pragmatic information about the associated turns. The second biggest corpus that is popularly used in this field of research is the ICSI Meeting Recorder Dialog Act (MRDA) Corpus [25], which includes over 180,000 hand-annotated dialogs and accompanying adjacency pair annotations for roughly 72 hours of speech from 75 naturally-occurring meetings among 53 speakers. There are three sets of dialog acts (DA) included: Basic (collapses all DA into 5 labels: Statement, BackChannel, Disruption, FloorGrabber, Question), General (uses 12 tags) and Full (uses all 52 tags).

A number of works have used a variety of AI inference models or statistical models to recognize and classify these fine-grained dialog acts with high accuracy. Formally, dialog act classification is the task of classifying an utterance with respect to the function it serves in a dialog. Notably among them, Colombo et al. [26] leveraged a seq2seq model using a hierarchical encoder, a novel guided attention mechanism and beam search for DA classification. Their proposed approach achieved an unmatched accuracy score of 85% on SWDA, and SOTA accuracy score of 91.6% on MRDA. Inspired by the observation that conversational utterances are usually associated with both a DA and a topic, Li et. al. [27] proposed a dual-attention hierarchical recurrent neural network for DA classification. The model reached an unmatched accuracy of 92.2% on MRDA and SOTA accuracy score of 82.3% on SWDA. Despite the high accuracy scores, DA recognition remains a challenging task to accomplish. This is because, although a few DA annotation schemes have emerged as standards, more often than not, DA tag-sets are extremely task-specific and need to be adapted accordingly. This prevents the deployment of standardized DA databases and evaluation procedures. For example, with the aim of facilitating the development of an automated dialog system for mental-health counselling, Malhotra et. al. [28] proposed a novel dataset called HOPE consisting of 12.9K utterances curated from publicly-available counselling session videos on YouTube. Given how different counselling and standard conversations are, the authors used 12 carefully designed dialog-act labels that are completely different from SWDA or MRDA in order to annotate their dataset. They also proposed a transformer based model (SPARTA) which achieved an accuracy rate of 64.75% on HOPE, surpassing all other baseline models. Likewise, with the goal of developing better learning environments and virtual mentors, Gautam et. al. [29] used 8 completely different DA labels to classify their dataset that comprises of conversations among students and mentors in Nephrotex (NTX), a virtual internship. Quinn et. al. [30] similarly looked into improving their conversational agent, ANA, by proposing 3 unique DAs: declarative, interrogative, and imperative to annotate their dataset because they fit into ANA's definition of a potential user utterance. These works further prove how different speech acts can be depending on the task/-domain at hand. As a result, concerns are raised about the generalization ability of a model trained on an unseen DA dataset even if it achieves SOTA performance on standard datasets like SWDA and MRDA.

Another challenge with dialog act classification is finding adequate training data annotated with domain-specific DA tags. This is a universal problem in NLP where most of the successful applications rely on supervised machine learning, which is notoriously data-hungry [31]. Generally speaking, the more labelled data is used to train a model, the better it gets. Obtaining this data especially for new domains or low resource languages, is often difficult, expensive, and time-consuming. As a solution to this issue, a number of researchers have looked into generating new data from existing ones. This is called data augmentation. More formally, data augmentation are techniques used to increase the amount of data by adding slightly modified copies of already existing data. The idea here is that the newly augmented dataset will be diverse enough to help smooth out machine learning models by reducing overfitting. Augmentation methods are quite popular in the domain of Computer Vision. Some of the widely used image augmentation algorithms include geometric transformations (rotation, flipping, cropping and scaling), colour augmentations (brightness, contrast, saturation and hue), kernel filters, random erasing and feature space augmentation [32]. Data augmentation has recently seen an increased interest in NLP due to more work in low-resource domains, new tasks, and the popularity of large-scale neural networks that require large amounts of training data. One popular augmentation technique is EDA or Easy Data Augmentation. It uses four simple approaches (synonym replacement, random insertion, random substitution, and random deletion) to alter the original texts and generate new ones [33]. Another technique called Back-translation leverages machine translation to paraphrase a text while retaining its meaning [34]. This is done by translating the original sentence to a target language and then translating it back to the source language. The idea here is that the re-translated text will be a bit different from the original one. More recently, Kumar et al. [35] proposed fine-tuning large pre-trained language models (BERT/GPT2/BART) to generate additional training data while preserving the class label. Upon using data augmentation, a number of researchers have reported improvement in model performance on a wide range of NLP tasks including text classification [36], summarization [37], and question-answering [38]. To address data scarcity and lack of diversity in DA classification task, researchers Kumar et al. [35] and Lai et al. [39] applied augmentation techniques like paraphrasing and substitution and reported improved model performance. Data augmentation was especially found useful in low-data regimes [35, 36]. However, such techniques must be chosen with extra caution, for the task of Sentiment Analysis, synonym replacement might work better than random deletion since the latter may change the meaning of the sentence entirely. Adding such a sentence to the training data that no longer preserves the original label may instead hamper the overall model performance.

Classifying the general user intent in a conversation, also known as dialog act, is a cruicial step in NLU for conversational agents. In this work, we propose a dialog act classifier for open-domain conversational agents. We first identify the relevant dialog acts for our existing chatbots (MIRA and ANA) and then curate a corresponding large-scale high-quality dataset of approximately 24K utterances. Each utterance of the dataset belongs to one of our eight proposed dialog acts, Statement, Factual Question, Yes/No Question, Direct Order, Indirect Order, Greetings, Feedback, and Apology. We later used an SVM classifier as a baseline and fine-tuned a pretrained BERT-base model for our DA classification task. Support Vector Machine (SVM) is a popular supervised learning algorithm used for classification problems. It works by creating a decision boundary that best segregates an n-dimensional space into classes. On the other hand, BERT is a transformer-based architecture which uses bidirectional training in order to have a deeper sense of language context. It simultaneously takes the previous and the next tokens into account and learns the text representations. Upon training, our BERT-based DA classifier achieves an accuracy of 99%, outperforming the baseline by 3%. We then address the problem of curating large-scale datasets by looking into a number of data augmentation techniques suitable for DA classification. By creating a low data regime (using only 10 examples per label), we show how data augmentation techniques as simple as synonym replacement can help the classifier yield an accuracy of 86%. Furthermore, on a small DA dataset translated to French, such methods can boost our model performance by 4%.

1.1 Thesis Statement

In this dissertation, we focus on building a classifier for open-domain conversational agents to accurately predict the dialog acts of users. This is done by addressing the two major issues related to DA classification: identifying DA tags suitable for our dialog system and dealing with scarcity of training data. To attain our objective, we first perform literature survey to identify suitable DA tags for two of our pre-existing chatbots (MIRA and ANA). We then automatically extracted subsequent examples for each tag from rich and diverse data sources to avoid mislabelling and to minimize human-error. The end product is a high-quality, large-scale DA dataset that spreads across multiple domains. We then fine-tuned the widely used state-of-the-art BERT-based model for our DA classification task. It outperformed our baseline SVM classifier and achieved a high accuracy on the proposed dataset. Given how difficult it is to gather massive amounts of annotated training data, we experimented with augmentation techniques suitable for DA classification and successfully improved model performance in low data regime. With the aim of building multilingual conversational agents, we also demonstrated how our classifier can be adopted to effectively detect dialog acts from French utterances.

1.2 Thesis Contributions

In this dissertation, we look into the use of data augmentation techniques when classifying dialog acts in conversational agents across multiple domains. In summary, the key contributions of this work and the new knowledge gained are as follows:

1. We proposed a taxonomy of 8 dialog acts suitable for our open-domain conversational agents and presented a corresponding high-quality, large-scale dataset of 24k user utterances,

2. We proposed a fine-tuned BERT-based model for the dialog act classification task and it outperformed our baseline SVM classifier by achieving a high accuracy,

3. We experimented with and compared a number of data augmentation techniques for different NLP tasks and showed how these methods can be successfully used to improve model performance for dialog act classification in low data regime,

4. We demonstrated how our proposed classifier and augmentation techniques can

be adapted to accurately detect dialog acts from non-English utterances like French.

1.3 Thesis Outline

The thesis is organized as follows:

In Chapter 2, we provide an overview of the datasets that are popularly used to train models for DA classification. We detail how these corpora were developed and the DA tag-sets that they used. Moreover, we introduce our proposed hierarchical taxonomy of 8 dialogue acts and present our corresponding dataset of 24k utterances. Details about data collection and preprocessing steps are also provided in this Chapter.

In Chapter 3, we provide a summary of the state-of-the-art DA classification models and include details about the model architecture. Next, we give a thorough background review of deep learning approaches for NLP. In particular, we start by explaining the basics of neural networks and then we move forward to discuss more advanced techniques like Recurrent Neural Network and Seq2Seq architectures.

Chapter 4 starts off by explaining in detail the architecture of our Bert-based classifier and SVM baseline for DA classification. Furthermore, we define several evaluation metrics for assessing the DA classifiers. Next, we exhaustively evaluate our machine learning models and analyze the results.

In Chapter 5, we provide a brief overview of some of the most prominent and the most recent augmentation techniques in the field of NLP. We then present a comparative analysis by applying a number of data augmentation techniques on a wide range of NLP tasks to show how much they can improve model performance.

Chapter 6 investigates the benefit of using data augmentation for DA classification in both English and French dataset. We first look into improving the performance of our baseline classifier by applying data augmentation for minority classes. Next, to fully utilize the potential of data augmentation, we create a low data regime and analyze the performance of our classifier with and without it. Lastly, we demonstrate how these techniques can be adapted to augment non-English datasets to improve our classifier performance on French utterances.

Finally, in Chapter 7, we summarize the results and the proposed contributions, address our limitations, and explore future works in the direction of dialog act classification for open-domain conversational agents.

Chapter 2

Dialog-Act Identification

Building conversational AI is a long-standing challenge in the field of NLP. Human conversations are inherently complex and ambiguous. Training a dialog system that understands the semantic and syntactic nuances and generates natural and engaging response is still difficult to achieve. However, recent works have shown the promise of combining dialog acts or speech intents for neural response generation [40]. Dialog acts can help conversational agents by providing a representation of the underlying meaning of a user's utterance. In order to drive the research on building better dialog systems, a number of conversational corpora have been released in the past. Such corpora usually consist of annotated dialogs in terms of speech or dialog acts that typically follow one of two paths [41]. The first path is task-oriented and categorizes speech acts for accomplishing a specific task in a domain. For example, the ATIS Airlines dataset which consists of audio recordings and corresponding manual transcripts about humans asking for flight information on automated airline travel inquiry systems [42]. The data consists of 17 unique intent categories related to inquiring about and/or booking flights. The second path aims for a more general coverage of day to day conversational dialogues. For example: the SWDA dataset which annotates speech acts for two-sided telephone conversations among 543 speakers from all areas of the US [43]. Given that both of our chatbots encourage open-ended conversations with users, our work mostly focuses on identifying dialog acts that follow the second path of research. This section gives an overview of some of the most prominent corpora used for classifying dialog acts in open-domain setting.

2.1 Related Works

Switchboard and MRDA are two of the most popular datasets used to train models for dialog-act detection. The Switchboard Dialog Act Corpus (SwDA) [43] extends the Switchboard-1 Telephone Speech Corpus, Release 2, with turn/utterance-level dialog-act tags [24]. The 43 tags used summarizes the syntactic, semantic, and pragmatic information about the associated turns. The SwDA project was undertaken at UC Boulder in the late 1990s with the goal of building better language models for automatic speech recognition. Given the minor distinction between some of the tags (like Statement-Opinion vs Statement-Non Opinion) and the fact that the dataset is highly unbalanced (Statement Non Opinion has 75145 utterances vs Thanking has only 78 utterances), it prompts a challenge to language models for proper identification of the dialog acts. Other notable tags in this corpus include Open-Question, Rhetorical Question, WH-Question, Yes/No Question, Apology, Summarize, and Appreciation. This corpus also has some tags that are unnecessary, confusing and/or not applicable for other domains (e.g., Non-verbal, Hedge, Conventional-closing, Collaborative Completion, Downplayer Maybe/Accept-part, etc.). Another famously used dataset for dialog intent detection is the Meeting Recorder Dialog Act Corpus (MRDA) [25]. The DA tagset for this dataset is just a modified version of the SWBD-DAMSL tagset. The whole dataset is annotated with three types of information: marking of the dialog act segment boundaries, marking of the dialog acts, and marking of correspondences between dialog acts. It consists of about 75 hours of speech from 75 naturally-occurring meetings among 53 speakers. As mentioned previously, the tags are subdivided into 3 sets: Basic (collapses all DA into 5 labels), General (collapses all DA into 12 labels), and Full (all 53 DAs). A language model can be trained to either identify only the basic/the general tags or all 53 tags, depending on the level of granularity required. Apart from SWDA and MRDA, a number of authors have also put forward DA annotated datasets for a wide range of tasks and domains. For example: Budzianowski et al. [44] proposed MultiWOZ- a Multi-Domain Wizard-of-Oz dataset which contains fully-labeled human-human written conversations spanning over multiple domains and topics. The proposed data-collection pipeline was based on crowd-sourcing and did not require hiring professional annotators. The authors show the usability of this dataset by presenting a set of benchmark results for belief tracking, dialog act and response generation. The 10k dialogues in this corpus are divided into 13 dialog acts, notable among which are inform, request, select, recommend, not found, welcome, greet and bye. Moreover, the corpus has dialogues that span over 7 common domains, such as Attraction, Hospital, Police, Hotel, Restaurant, Taxi and Train. With the aim of improving the research field of dialog systems, Li et al. [45] proposed DailyDialog, a high-quality multi-turn dialog dataset of approx. 13k utterances which resembles casual conversations that human beings have in their day to day life. The largest three categories of conversations include Relationship (33.33%), Ordinary Life (28.26%), and Work (14.49%). The dataset is less noisy and is manually annotated with dialog intents and emotion information. The 4 dialog intents are very straightforward and follow the criteria proposed by Amanova et al. [46]: Inform (includes statements and questions that provide information), Questions (includes dialogues that seek information), Directives (includes acts like request, instruct, suggest and accept/reject offer), Commissive (includes acts like accept/reject request or suggestion and offer). With the aim of jointly modelling mutual relationships and integrating intention information, Peng et al. [47] extracted 2046 conversations from DailyDialogue and classified the dialog acts into seven categories: request, suggest, command, accept, reject, question and inform. Their qualitative analysis verified the importance of mutual interaction between intention and emotion. As user intents change with the change in domain/topic, proposed schema for dialog acts need to be adjusted accordingly. This is inconvenient when it comes to training a dialog system that generalizes well on multiple domains. To tackle this issue, Paul et al. [48] proposed a Universal DA schema for task-oriented dialogs to allow training a Universal DA tagger for a multitude of dialogs. Their proposed schema has 20 universal DA tags, including acknowledge, affirm, bye, deny, inform, repeat, request, restart, thank-you, and user-confirm. To make the transition easier, they also propose manual and automated approaches for aligning the different schema.

More recently, there has been a rise in the number of conversational agents designed to provide social and emotional support in healthcare and personal lives [49]. With the intention of progressing the development of automated dialog system for mental-health counselling, Malhotra et al. [28] proposed HOPE, a novel dataset curated for dialogue-act classification in counselling conversations. The authors proposed 12 domain specific dialogue-act labels based on the data they gathered (transcripts of counselling session videos on YouTube). The 12 DA labels are separated into 3 groups, Speaker initiative labels (Information Request, Yes-No Question), Speaker responsive labels (Information Delivery, Positive Answer), and General (Greeting, General Chit-Chat). The authors also proposed SPARTA, a novel transformer-based speaker and time-aware joint contextual learning model for DA classification on the HOPE dataset. Similarly, Welivita et al. [40] proposed a large-scale taxonomy for empathetic response intents in order to help open-domain conversational agents generate socially acceptable responses. The authors manually labeled 500 utterances from an empathetic dialog dataset with 15 Empathetic Response Tags, such as Consoling, Encouraging, Sympathizing, Wishing, Suggesting, Advising, Expressing care or concern, and Expressing relief. They hope that the dataset will help chatbots generate prosocial conversations that are engaging and empathetic to users. Saha et al. [50] emphasize the influence of non-verbal cues and emotional state of speaker when classifying dialog acts and propose a new multimodal Emotion aware Dialog Act dataset called EMOTyDA. It was curated with utterances from the MELD [51] and the IEMOCAP [52] dataset and used 12 of the 42 intents introduced in the MRDA dataset. Like the previous authors, they also built an attention-based multi-modal, multi-task Deep Neural Network for joint learning of DAs and emotions. Quinn et al. [30] similarly proposed 3 dialog acts: Declarative, Interrogative, and Imperative for their chatbot, ANA, which communicates with the elderly and tries to improve their quality of life.

A number of conversational agents are also used for online educational games, courses and interactive virtual internships. As such, the dialog intents for this domain are very different from the ones discussed previously. Gautam et al. [29] proposed 8 speech acts after analyzing 22k chat utterances in Nephrotex virtual internship in order to better understand the actions or intents behind each utterance in the conversations between interns and mentors. Notable DA tags include expressive evaluation, greeting, question, reaction and request. Samei et al. [53] looked into classifying speech acts in intelligent tutoring systems in order to inform the system's response mechanism. The authors extracted 26K mentor-student chat utterances from seven Land Science games and adjusted the training data to include an even distribution of 30 instances per speech act category. The categories include: Statement, Question, Reaction, Request, MetaStatements, Greetings, and ExpressiveEvaluation. Upon experimentation, they discovered the importance of context in automatically predicting speech acts. Arguello et al. [54] did something similar with massive open online courses (MOOCs) where students interact with each other and the course staff through online discussion forums. They categorized 2943 individual messages into seven speech acts (question, answer, issue, issue resolution, positive and negative acknowledgment, other) to help instructors answer questions, resolve issues, and provide feedback more efficiently.

Several researchers have also focused on detecting speech acts in emails and online meetings. For example, the AMI meeting corpus [55] is a multi-modal dataset consisting of 100 hours of meeting recordings annotated with 15 DAs that involve special classes to allow complete segmentation, information exchange, possible actions, and social acts. Cohen et al. similarly [56] presented an ontology of ‘email speech acts’ to capture important properties of emails like negotiating and coordinating joint activities. They classified 15K fake emails sent back and forth between MBA students into 7 speech acts: Request, Deliver, Commit, Propose, Directive, Commisive, and Meeting. Embar et al. [57] focus on extracting action items from online meetings and classified them into 4 categories: Commitments, Directives, Acknowledgement, and Elaborate. They further split Directive and Commitment tags into in-meeting and post-meeting based on whether those were resolved during the meeting or need to be resolved afterwards. They trained the RoBERTa model on the annotated dataset of 50,000 sentences and achieved an accuracy of 82%.

Some researchers have also looked into classifying speech acts of conversations from social websites like Twitter and Stack Exchange. Vosoughi et al. [58] considered recognising speech acts of Tweets by treating it as a multi-class classification problem. The authors collected a total of 7563 Tweets relating to 6 topics and 3 types and using Searle’s speech act taxonomy [59], they established a list of 6 speech act categories: Assertion, Recommendation, Expression, Question, Request, and Miscellaneous. Similarly, Zhang et al. [60] manually annotated a total of 8613 tweets as Statement, Question, Suggestion, Comment, or Miscellaneous. Using a set of word-based and character-based features, their model achieved an average F1 score of nearly 0.70 on the dataset. With the goal of building more generalized conversational models, Penha et al. [61] proposed a large-scale multi-domain information seeking dialog dataset. They used a total of 9 DA categories for almost 80K conversations across 14 domains extracted from Stack Exchange. The conversations are either questions (Original question, Follow Up Question, and Information Request), answers (Potential Answer, and Further Details), gratitude (Greeting/Gratitude) or feedback (Positive Feedback, and Negative Feedback). Their proposed DA taxonomy was initially coined by Qu et al. [62] for their MSDialog dataset consisting of 10,000 utterances from an online forum on Microsoft products. Wood et al. [63] was more specific and addressed the problem of speech act detection in conversations about bug repair. They conducted a "Wizard of Oz" experiment with 30 professional programmers and tasked them with solving bugs for two hours while using a simulated virtual assistant for help. Upon analysing 30 two-hour conversations, they uncovered 26 speech act types like syntaxQuestion, parameterQuestion, unsureAnswer, confirmation, instruction and more.

Most of the proposed DA tagsets so far rely upon human-human conversations for training dialog systems. Given how differently humans interact with other humans vs. with machines, Yu et al. [64] proposed a dialog act annotation scheme called MIDAS based on human-machine conversations in open domain setting. MIDAS supports multi-label annotations, provides context completeness and follows a tree structure. Figure 2.1 and 2.2 show how the 23 DAs are distributed under the two sub trees- semantic request and functional request. The authors also collected and annotated 24k segmented sentences with their proposed schema and used transfer learning to train a multi-label dialog act prediction model on it which achieved an F1-Score of 0.79.

Like the previous works, this thesis aims towards building conversational agents that respond naturally through accurate detection of user DA. In particular, we focus on building a DA classifier that is applicable for our pre-existing text-based chatbots- ANA and MIRA. Apart from answering questions and sending reminders, Automated Nursing Agent or ANA aims to have a fluent and personalized conversation with the elderlies [30]. On the other hand, MIRA is a Mental Health Virtual Assistant which provides mental health resources to health care workers [65]. It also has a module called 'Chatty MIRA' which allows users to have open-ended conversations with the chatbot. Given the difference in domain and task intents, our goal is to propose a dialog act schema that is common to both. The next section gives a more detailed explanation on how the DA tagset was chosen.

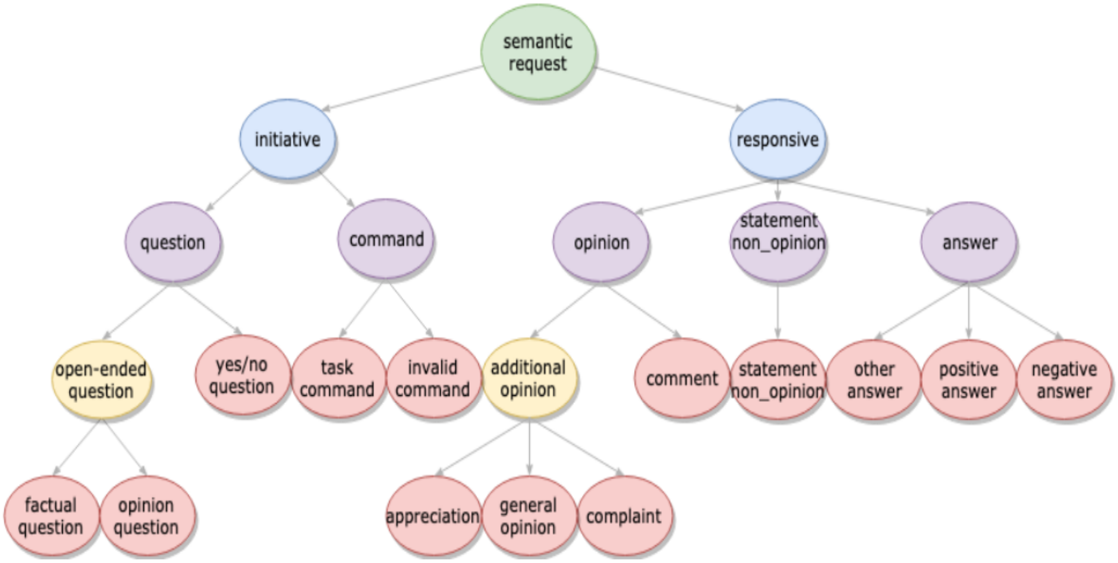


Figure 2.1: Semantic request tree. Scheme types, classes, categories, sub-categories and dialog act tags are in green, blue, purple, yellow, and red respectively. Most tags can co-occur in one utterance (exception: tags under opinion, statement non opinion, question, and answer categories) [64]

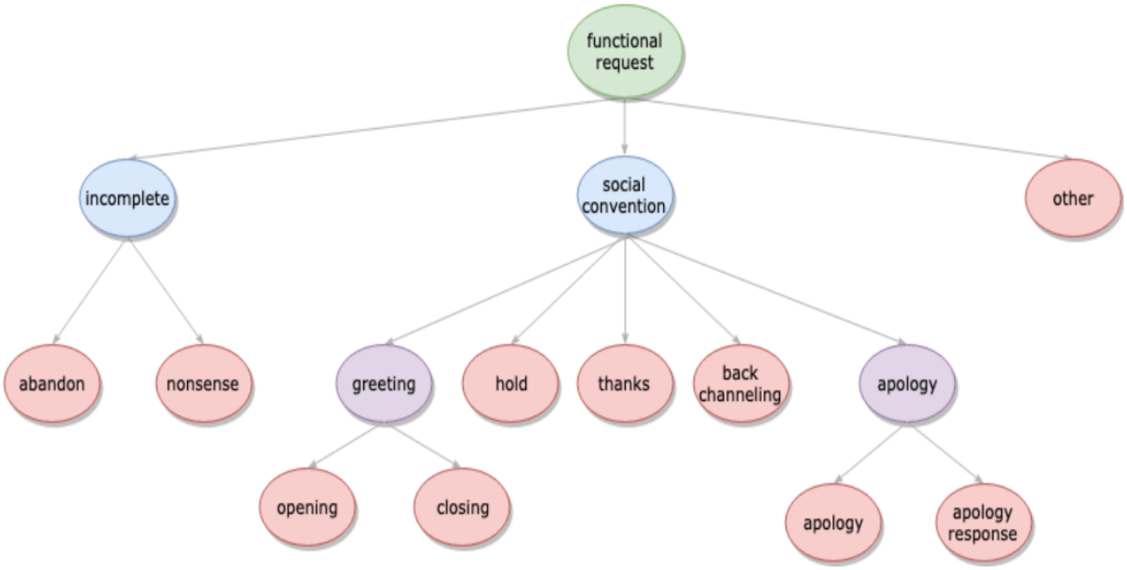


Figure 2.2: Functional request tree [64]

2.2 Initial Taxonomy (Three Dialog Acts)

The first step towards building a DA classifier for a conversational agent is identifying the different dialog acts a user may perform. For this, we reviewed the existing literature on dialog systems to identify DAs that are common and might be suitable for our chatbots. Accordingly, we recognized a number of frequently occurring DAs such as Question (Yes/No, Factual, Tag, WH), Statement (Opinion, Non-Opinion), Greeting, Apology, Command, Request, Gratitude, Accept, Reject, Wish, and Suggest, and took them into consideration. However, for the purpose of simplification, we initially took inspiration from the work by Quinn et al. [30] where they had divided the dialog acts into 3 broad categories: Interrogative (e.g., What's the weather like today?), Imperative (e.g., Call my doctor) and Declarative (e.g., I am 57 years old). Although straight-forward, this taxonomy is inclusive and can easily classify any user utterance under one of the three categories. This also makes the DA tagset generalized enough to be applied to any domains and easy enough to be aligned with schema from previous works. With all this in mind, we decided to move forward with these three dialog acts for both of our chatbots.

The next step towards building a dialog act classifier for our chatbots is curating a training dataset based on the proposed DA tagset. Intuitively, we decided to use the dataset proposed by Quinn et al. [30] for training our model. However, this was not possible because, although the authors had mentioned extracting their data from a number of websites such as ‘uselessfacts.net’, ‘wikihow.com’ etc., the final training dataset was too small, with only around 1500 sentences. When trained on such an insufficient amount of data, their SVM classifier performed poorly, with a reported accuracy of only 82%. Moreover, upon manual investigation, we found out that the dataset was very noisy with many mislabelled sentences. As a result, we decided to extend their work by addressing the limitations and creating a new dataset that has a large number of examples for each label (Interrogative, Declarative/Statement, and

Dataset	Dialog Act	Train Data	Test Data	Distribution	Example
Ours	Imperative (I)	9825	2456	27%	Complete the photo shoot.
Kevin et al		367	88	31%	Throw me the ball.
Ours	Question (Q)	11629	2908	31%	What are the income taxes in Canada?
Kevin et al		237	61	20%	So when are you leaving camp?
Ours	Statement/Declarative (S)	15645	3911	42%	Residents of Kansas are called Kansans.
Quinn et al		653	73	49%	More people are killed each year from bees than from snakes.

Table 2.1: Overview of our proposed training and test dataset

Imperative) and little to no mislabels. The hope is that a larger, cleaner, and more diverse training dataset will help improve model performance. Last but not the least, for further yield in accuracy, we train a state-of-the-art BERT-base model for DA classification and compare it with the performance of a classic SVM classifier.

2.2.1 Data Source

We extensively looked into a number of pre-existing datasets and websites and used multiple rules to extract suitable examples for each of our dialog acts. Table 2.1 provides details on the number of training and test samples we gathered per class along with an example. Below, we list down all the data sources we had used for each DA:

1. Statement: We curated training and test examples from sources that are most likely to contain statements/opinions. Examples include Wikipedia Articles, Daily Dialogue [45], Amazon Product Reviews, and IMDB Movie Reviews [66]. During extraction, we made sure the extracted sentences followed the common subject + verb + object structure ('She played the role of Annie') or a variation of it ('I am trying very hard.'). More importantly, we ensured that these sentences do not follow the rules that we had used to identify questions and imperatives.

2. Imperative: We studied a number of literature to get an idea of the rules that can be used to accurately identify imperatives. Abdul-Kader et al. [67] in their work only considered sentences that begin with the base-form of a verb

(e.g., ‘Play music’). But this leaves out a good portion of imperative sentences that do not follow this rule, such as ‘Hey Alexa, please tell me a joke’. Mao et al. [68] in their work, included two more rules that are useful in detecting imperatives. One of them being identifying sentences that have a verb in its lemma (base) form and is the root and does not have any subject child in its dependency structure, ‘Just practice programming thoroughly’. Another rule involved recognizing the use of a personal pronoun (you) followed by a modal verb (should, must, and need to) as an imperative, ‘You should study hard.’ After merging these rules, we identified and extracted training and test examples from websites and pre-existing datasets Friends TV series subtitles, Reddit posts, Tweets, Movie Subtitles, and HowTo datasets. However, due to the limitation of NLP libraries like Spacy, these rules sometimes tag incorrect sentences as imperatives, so tweaks to the rules were made as per the dataset in hand.

3. Interrogative: For extracting questions/interrogative sentences, we mostly leveraged pre-existing datasets that are widely used for NLP question-answering tasks, including SQUAD [69], and TREC[70].

In the end we were able to curate a training and test dataset with a total of 37099 and 9275 examples. To put into perspective, our dataset is approximately 31 times larger than Quinn et al. [30]. Given the size of the dataset and the number of rules used to extract them, misclassification of examples during extraction is a legitimate concern. In order to ensure that the extracted dataset is clean and contains minimal misclassification, we took the help of 5 annotators. The annotators were first provided with the definition of the three dialog acts and then each of them were given 100 random examples from each label and were asked to check whether the sentences belonged to the corresponding class. In case of disagreement between annotations, it was resolved through open discussion that led to the re-evaluation of the definitions and coming to a unanimous decision. The experiment was repeated 20 times and on an average, the data was found to be 91% clean (i.e., only 9% of the data was mislabelled).

One important thing to note is that, all the sentences in our dataset are stripped of punctuation marks. This is because most of our users do not use proper punctuation marks when chatting with our chatbots. Without punctuation, the task of identification becomes much more difficult for the classifier. This is because usually if a sentence had ended with a question mark, we could assume it is a question. For the sake of simplicity, like Quin et al., we assume a sentence can only belong to one class. However, this is not always true in practice. For example, the sentence 'Do you have the time?' can be a request (imperative) or question (interrogative).

We used it to train the SVM baseline as well as our pretrained BERT-based classifier and evaluated their performance. Both achieved high accuracies (well above 90%), which surpassed that of Quinn et al. More details about the experimental set up and the results are provided in Chapter 4.

2.3 Proposed Taxonomy (Eight Dialog Acts)

Our newly curated large-scale dataset with three dialog acts (Interrogative, Imperative, and Declarative) fulfilled our initial goal and improved the performance of both of our DA classifiers. However, we soon realised that our DA tagset was too general and failed to capture a number of cases that require different responses from the chatbot. For example, 'Can penguins fly?' and 'What is the name of our galaxy?' are both questions. However, the first one expects a yes/no answer from the chatbot, whereas the second one expects a factual answer. In order to generate better responses, our chatbots need to learn how to distinguish between the two. In other words, our proposed DA tagset needs to be more inclusive to handle user acts like this. Another limitation of our dataset revolves around the data sources themselves. Upon analysis of the examples for each label, we soon realised that a lot of the sentences were not reflective of the type of conversations a user would have with our chatbots. For example, 'Don't go out in the sun' is an imperative sentence but it is not an order a user would give to a chatbot. Similar is the case with including too many sentences from Wikipedia or news articles. Although these are convenient sources of Declarative sentences, the statements are often too long, formal, and not representative of how human beings converse. To make our chatbots more capable of recognising user intents, it is necessary to use data sources that have utterances that a user is more likely to use when chatting with a conversational agent. Taking all these into consideration, we decided to expand our previous taxonomy and change our data sources. For this, we first investigated the existing literature and identified the dialog acts that are common in multiple domains. Next, we looked into the chat history between our users and chatbots in order to have an idea of the types of dialog acts to consider. Finally, we had iterative discussions with our cross-functional team comprising of people from Computer Science, Health Science, and Psychology. In the end, we selected the following eight dialog acts and categorized them into a hierarchy for a better understanding. We may extend the number of dialog acts in the future if deemed necessary. Table 2.2 provides examples for each of the dialog acts and the later sections talk in depth about our new sources of data.

1. Apology: Includes sentences through which the user expresses apology.

2. Greeting: Includes sentences through which the user greets the chatbot either at the beginning or towards the end of a session.

3. Informative: Includes queries asked by the user with the intention of gaining some information. Depending on the type of response, questions can broadly be of 2 types:

    (a) Yes/No: Includes close-ended queries that can be sufficiently answered with a simple yes or no.

Dialog Act	Sub Category	Example
Apology		Sorry about that!
		My bad.
Greeting		Hey, how are you?
		Bye, see you soon!
Informative	Yes/No Question	Is it possible to treat ADHD?
	Factual Question	What year did Bangladesh achieve their independence?
		Name the best therapist in my area.
Directive	Direct Order	Show me the list of hospitals nearby.
	Indirect Order	I need help with managing anxiety.
		Can you turn on the music please?
Statement		I am being bullied at school lately.
		I like spending time with my family.
Feedback		This is exactly what I was looking for! Thanks.
		This not what I wanted. You suck!

Table 2.2: Selected dialog acts with examples

(b) Factual: Includes open-ended queries that seek fact-based answers. A majority of these questions are WH-questions but utterances like 'Name the highest rated therapist in my area' are also included here due to the similarity in user intent.

4. Directive: Includes orders given by the user to the chatbot for accomplishing a task. This again can be of two types:

   (a) Direct Order: Includes straightforward orders that are easy to detect, understand and carry out.

   (b) Indirect Order: Includes utterances that indirectly expect or request some type of action. These are a bit difficult to comprehend and might require the chatbot to first make an assumption and then prompt for a confirmation before execution. For example: ‘I need help with managing anxiety’ or ‘Can you help me manage my anxiety?’ can be interpreted as ‘Show me resources for managing anxiety’.

5. Statement: Includes user utterances that do not request for an action or information. Rather, these are dialogs through which the user casually converses with the chatbot. By analyzing the emotion behind these utterances, the chatbot can either choose to give a sympathetic response or ask follow-up questions.

6. Feedback: Includes feedback from the user once the chatbot accomplishes a task (e.g., carries out an order or answers a question). Feedback can be positive (when the chatbot is successful) or negative (when the chatbot is unsuccessful). By detecting the sentiment behind it, the chatbot can either thank the user or apologize and/or attempt the task again.

2.4 Data Source

Since we plan on developing a DA classifier applicable for both of our chatbots, we need our training dataset to be both versatile and general. For this, we decided to include examples that are not only related to mental health (for MIRA) and popular chatbot commands (for ANA), but also common domains like banking, air lines, or product reviews etc. The idea here is that the difference in user intents will add variation in structure and make the dataset more diverse. As for extracting the examples themselves, we first looked at the popular DA datasets that are available online. Some of them had examples for tags that are similar to ours whereas for the rest, we had to scrap various websites and forums. It is to be noted that, during data collection, we decided to include only those examples that followed our definition of each of the dialog acts. To avoid dominance of a particular domain or type of sentence, we made sure not to include too many examples belonging to the same

Dialog Act	Sub Category	Train Examples	Test Examples	% Distribution
Informative	Yes/No	3385	847	17.31
	Factual	3697	924	18.89
Directive	Direct Orders	3125	781	15.97
	Indirect Orders	5400	1349	27.60
Statement		3250	812	16.61
Greetings		239	60	1.22
Feedback		392	98	2
Apology		73	18	0.4

Table 2.3: Overview of our proposed training and test dataset

domain or sentence structure. A detailed description of each of the data sources that were used has been provided in Table 2.4. Below, we give a brief overview of that:

1. Informative: We mostly used popular question-answering datasets, a few task-completion datasets and mental health FAQ websites.

   (a) Yes/No Question: BoolQ [71], SNIPS [72]

   (b) Factual Question: SNIPS[72], SQUAD [69]

2. Directive: We used task-completion dialog intent datasets to collect Direct and Indirect Orders. Simple extraction rules were used to distinguish between the two. Datasets include Taskmaster [73], SNIPS [72], ATIS [42] and ACID [74] to name a few.

3. Statement: We mostly used the dataset shared by a mental health forum called ‘Counsel-Chat’, which consists of anonymous user posts related to mental health. We also included some examples from Wiki-Article, IMDB Movie Review and Amazon Product Review datasets.

4. Feedback, Apology, and Greeting: We scraped a few basic English learning websites to extract positive and negative appraisals, apologies and greetings.

Dataset	Description	Used In	Examples
BoolQ [71]	A question answering dataset for yes/no questions containing 15942 examples. The questions are naturally occurring and were generated in unprompted and unconstrained settings	Yes/No	Do iran and afghanistan speak the same language?
SNIPS [72]	A dataset of over 16,000 crowdsourced queries distributed among 7 task-oriented user intents of various complexity (SearchCreativeWork, GetWeather, BookRestaurant, PlayMusic, AddToPlaylist, RateBook, SearchScreeningEvent)	Direct Order Indirect Order Factual Yes/No	Find me the I, Robot television show I want to book a highly rated restaurant in Paris tomorrow night What's the best hotel between Soho Grand and Paramount Hotel? Is my Airbnb closer than John's hotel?
SQuAD [69]	A reading comprehension dataset consisting of questions posed by crowdworkers on a set of Wikipedia articles. The answer to every question is a segment of text, or span, from the corresponding reading passage. There are 100,000 question-answer pairs on 500 articles	Factual	What is the daily student paper at Notre Dame called?
Mental Health FAQ for Chatbot	A Kaggle dataset consisting of 98 FAQs about Mental Health by scrapping mental health websites and forums	Factual	What does it mean to have a mental illness?
ACID[74]	Contains 174 intents collected from customer interactions with the service representatives at American Family Insurance. Each intent represents a particular course of action for the chatbot. The training set has 11130 examples and test set has 11042 examples	Direct Order Indirect Order	Please remove my son from my auto policy and create a separate one for him. Can you take my son off of my car insurance policy, please ?
CLINC150	A complex intent detection dataset with two separate domains- 'Banking' and 'Credit cards' with both general and In-Domain Out-of-Scope queries. Each domain originally includes 15 intents.	Direct Order Indirect Order	Tell me how to set up a direct deposit Can you show me how to set up direct deposit for paycheck to my bank account?
ATIS[42]	A dataset consisting of audio recordings and corresponding manual transcripts about humans asking for flight information on automated airline travel inquiry systems. It has 17 unique intents. The original split contains 4478, 500 and 893 intent-labeled reference utterances in train, development and test set	Direct Order Indirect Order	Show me the flights from Pittsburgh to Los Angeles on Thursday I need a flight tomorrow from Columbus to Minneapolis
Taskmaster [73]	Consists of three datasets, Taskmaster-1, 2 and 3, comprising over 55000 spoken and written task-oriented dialogues in over a dozen domains (ordering pizza, setting up ride service, ordering movie tickets etc.). Two procedures were used to create this collection: the first involves a two-person, spoken 'Wizard of Oz' (WOz) approach while the second is a 'self-dialog' approach	Direct Order Indirect Order	Show me the movies for Boston, Massachusetts. Hi, I would like to buy 2 tickets for Shazam!
Counsel-Chat	A mental health dataset shared by the founders of Counselchat.com, a platform in which therapists respond to questions posed by clients, and users can like responses that they find the most helpful	Statement Factual Indirect Order	My wife and mother are having tense disagreements. What can I do to get rid of this addiction? I need help with issues of abuse as a child, addiction, and abusive men.
WikiArticle	A multimodal dataset of 'good articles' on Wikipedia containing 36,476 articles and 216,463 images available on Kaggle. It contains the text of an article and also all the images from that article along with metadata such as image titles and descriptions. The selected good articles are just a small subset of the available ones, but have been manually reviewed and protected from edits.	Statement	Cooper eventually deduces the patterns were caused by gravity variations and are a binary code for geographic coordinates.
IMDB Movie Review [66]	A binary sentiment analysis dataset consisting of 50000 reviews from the Internet Movie Database (IMDb) labeled as positive or negative. The dataset contains an even number of positive and negative reviews. No more than 30 reviews are included per movie.	Statement	This film is just plain horrible.
Amazon Product Review	A dataset to tackle the task of identifying the sentiment of a product review (positive or negative). It includes reviews from four different merchandise categories: Books (2834 samples), DVDs (1199 samples), Electronics (1883 samples), and Kitchen and housewares (1755 samples).	Statement	However, when I purchased this product from trade concepts, I received the worst espresso I have ever tasted.
Smalltalk	A dataset of casual conversations that was collected to be used with the Rasa Stack. It includes examples for intents like greetings, good appraisal, bad appraisal etc.	Greetings Feedback	How's your day going? You helped a lot thank you

Table 2.4: Data sources with description and types of examples used

Table 2.3 shows how the examples are distributed per class. Our dataset initially consisted of 24,450 examples in total. We divide it to create a train and a test dataset. The split is done in a way to include 25% of the examples for each class in the test dataset. This was done to tackle the imbalance that exists with the minority classes (Apology, Greeting, and Feedback). For Yes/No Questions, more examples were taken from BoolQ [71] and a few from SNIPS [72] dataset. For Factual Questions, apart from the 90+ mental health related questions collected from different websites, the rest were taken from SQUAD [69] and SNIPS dataset [72]. For Direct and Indirect Order, we included many examples from Taskmaster [73] and SNIPS [72] dataset that fit our definition of the dialog acts. A few were extracted from the ACID [74], CLINC150, and ATIS [42] datasets. We also scrapped different websites to obtain a list of common imperative sentences as well as popular commands/requests given to Alexa, Siri, and Google Assistant. For Statement class, we included most of the examples from Counsel-Chat. This was done for two main reasons. Firstly, because it pertains to mental health and, hence, it would require the chatbot to respond more emotionally and sympathetically. Secondly, because this is how we assume our users would talk to our chatbots, by opening up about day to day issues that are bothering them. This makes the newly curated dataset more authentic and realistic. We also made sure to included a few examples from product and movie reviews as well as wiki articles to add some variety to the structure of declarative sentences. As for minority classes like Apology, Greeting and Feedback, the Smalltalk dataset was mostly used. Due to the lack of variation in the ways users can greet, apologize and provide feedback in real life, these three classes have a small number of examples in comparison. Finally, it is important to mention that, because a majority of examples for each class was taken from popular datasets, the chance of including mislabelled sentences in our final dataset is minimal. This makes our proposed DA dataset extremely reliable and free of errors.

Chapter 3

Background

The task of predicting user dialog acts (DAC) during a conversation is a key component in building a conversational agent. As a result, a number of researchers have proposed a wide variety of deep learning models for classifying dialog acts and have tested their performance on some of the most popular DAC datasets such as SWDA and, MRDA. In the following section, we will provide a brief overview of some of these influential works.

3.1 Dialog Act Classification

A number of recent works have treated the task of dialog act classification as a sequence labeling problem. Colombo et al. [26] in their work leveraged a seq2seq model using a hierarchical encoder, a novel guided attention mechanism and beam search applied to both training and inference. Seq2seq models are widely used in NMT and are popular for learning complex global dependencies. Unlike other models, theirs does not require hand-crafted features for training. The proposed model achieved an unmatched accuracy of 85% on SwDA and 91.6% on MRDA. Raheja et al. [75] used an amalgamation of self-attention, hierarchical deep learning models and contextual dependencies to build their DAC classifier. They proved through their experiments that utterance representations learned at lower levels impact the classification performance at higher levels. With the help of self-attention, their model is able to learn richer, more effective utterance representations and achieve an accuracy of 82.9% on SWDA and 91.1% on MRDA. Inspired by the observation that conversational utterances are normally associated with both a DAC and a topic, Li et al. [27] proposed a dual-attention hierarchical recurrent neural network with a CRF (Conditional Random Field) for DAC classification. The novel dual task-specific attention mechanism helps their model capture information about not only the DACs and topics but also the interactions between them. They achieved an accuracy of 82.3%, 92.2%, and 88.1% on SWDA, MRDA, and Daily Dialog dataset, respectively.

Apart from SWDA and MRDA, a number of researchers have also proposed DAC classifiers for other pre-existing as well as self-curated domain-specific datasets. For example, Ahmadvand et al. [76] proposed a novel contextual DAC classifier that uses transfer learning to adapt models trained on human-human conversations to predict dialog acts in human-machine dialogs. The model incorporates lexical, syntactic, and semantic information as context and was evaluated by first training it on the SWDA human-human dialog dataset, and later fine-tuning it for predicting DACs in human-machine conversation data, collected as part of the Amazon Alexa Prize 2018 competition. With the recent success of BERT (Bidirectional Encoder Representations from Transformers), Saha et al. [77] proposed BERT-Caps, a BERT-based model that learns traits and attributes by leveraging from the joint optimization of features from the BERT and capsule layer. Their proposed model attained a benchmark accuracy of 77.5%, outperforming several strong baselines and state-of-the-art approaches for dialog act detection in tweets. Similarly, Wu et al. [78] proposed a context-aware hierarchical BERT fusion Network for DAC classification in multi-turn dialogs. Their proposed model not only discern context information within a dialog but also jointly identify multiple DACs and slots in each utterance. As a result, it was able to outperformed previous Spoken Language Understanding models that only consider single utterances for multiple intents and slot filling in two complicated multi-turn dialog datasets, Microsoft dialog Challenge dataset [79] and Schema-Guided dialog dataset [80]. A number of research works [81, 82] have also looked into incorporating emotion/sentiment of texts as context during DAC classification. Qin et al. [82] proposed a Deep Co-Interactive Relation Network (DCR-Net) that explicitly models the interaction between the two tasks: dialog act recognition and sentiment classification by introducing a co-interactive relation layer. Their model outperformed the SOTA model on Mastodon [83] and Daily Dialog [45] dataset for both the tasks. They also incorporated BERT in their framework to boost performance. Huber et al. [84] performed DAC classification in a unique domain (Parent-Child Interaction Therapy) to inform parents about their DAC use. For this, they first created a dataset of 6,022 parent utterances that were annotated by experts with dialog act labels that therapists use to code parent speech. Next, they developed an algorithm that classified the dialog acts into eight classes with an overall accuracy of 78%. Quinn et al. [30] also curated their very own dataset with three DACs and utilized an SVM classifier to achieve an accuracy rate of 82%.

These findings suggest that, when it comes to DAC classification, researchers propose their models based on the dataset at hand and incorporate different features and complexities to improve performance. The rest of the chapter is dedicated towards reviewing the technical background of NLP based deep learning models as well as exploring the existing dialog systems.

3.2 Artificial Neural Networks (ANN)

An artificial neural network is a computer system modeled around how the human brain and nervous system functions. ANNs learn tasks automatically by looking into examples without being explicitly programmed. They do so by deriving meaning from unstructured data and by capturing high-level representations that are considered too complex for either humans or other computer techniques. Neural networks were first proposed in 1944 by Warren McCullough and Walter Pitts [85]. A typical neural network has neurons, often called units or nodes.

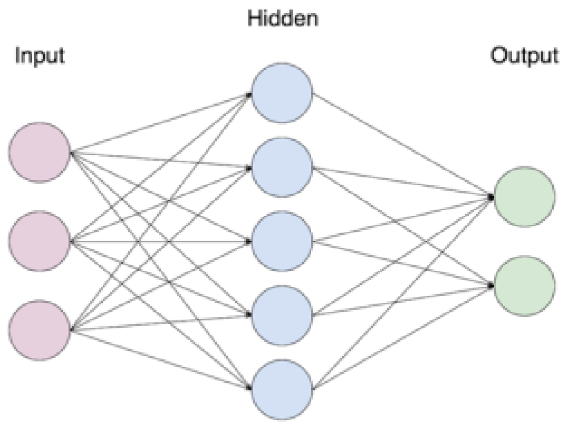


Figure 3.1: Workflow in Artificial Neural Networks [86]

Figure 3.1 shows the input layer consisting of input units that receive numerical data from outside. The connection between one unit from input layer and one from hidden layer is represented by a number called a weight. The weight can be either positive or negative; corresponding to the way actual brain cells excite or suppress others. If a unit has a higher corresponding weight, it has more influence on the output. Although assigned at random initially, the weights are later adjusted through the training process. McCulloch-Pitts introduced a simplified model of the human neuron as a mathematical linear function that receives a set of n inputs $x_1, ..., x_n$ and linearly transforms them to an output y. This model learns a set of weights w_1, ..., w_n and calculates the output $y = f(x, w) = x_1w_1 + ... + x_nw_n$. Their neuron predicts two different groups of inputs by corresponding to whether $f(x, w)$ is positive or negative.

3.2.1 A Single Neuron

Figure 3.2 shows a network consisting of one hidden layer containing one neuron. The single neuron receives input from the prior input layer, performs some computation and sends the result away. The neuron has two inputs, x_1 and x_2, with weights w_1 and w_2, respectively. The neuron applies a function f to the dot-product of these inputs, which is $w_1x_1 + w_2x_2 + b$. Besides the two numerical input values, there is one input value 1 with weight b, called the bias. It stands for unknown parameters

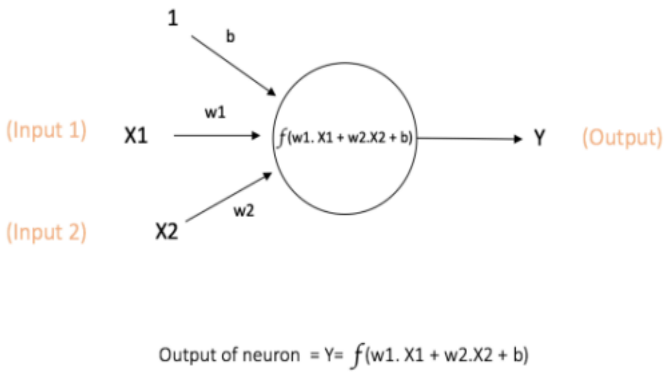


Figure 3.2: A single neuron in neural networks [86]

or unforeseen factors. The output Y is computed by taking the dot-product of all input values and their associated weights into the Activation Function, f. It is used to produce results in desired ranges (between 0 to 1 or -1 to 1 etc.). Some frequently used activation functions include:

1. Sigmoid or Logistic: takes a real-valued input and returns an output in the range [0,1]: $\delta(x) = \frac{1}{1+e^{-x}}$

2. tanh or hyperbolic tangent: takes real-valued input and produces the results in the range [-1, 1]: $tanh(x) = \frac{sinh(x)}{cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

3. ReLU (Rectified Linear Unit): takes a real-valued input and replaces the negative values with zero: $R(x) = max(0, x)$

3.2.2 Feed-Forward NN

A feed-forward neural network is a multi-layer network with three layers: input, hidden and output. It works by feeding the outputs from neurons in one layer to the next layer. It is a fully connected network in which each layer takes all the outputs from the previous layer as input. Also, there is no link connecting units in the same layer. The input layer units are scalar values while the units in the hidden layer correspond to neural units. These neural units compute a weighted sum of their inputs and then apply a non-linear activation function like tanh or sigmoid. Formally, the output of the hidden layer is as follows:

$$h = f(wx + b)$$

where f = a non-linear activation function, $x \epsilon R^{d_{in}}$ = a vector of real numbers representing the inputs, d_{in} = the number of inputs; $b \epsilon R^{d_h}$ = the bias and $W \epsilon R^{d_h x d_{in}}$ = the weight matrix.

Finally, the output layer computes a final output based on the representation value $h \epsilon R^{d_h}$. Depending on the task at hand, this value can be a real number or probability distribution across the vocabulary words. Like the hidden layer, the output layer also has a weight matrix U and often does not have a bias vector. The network multiply weight matrix U by the hidden vector h to generate an output z as follows:

$$z = Uh$$

where $z \epsilon^{d_{out}}$ with d_{out} is the number of output units and $U \epsilon R^{d_{out} x d_h}$. For classification tasks, the output should be a normalized vector of probabilities (vectors that range between 0 and 1 and sum to 1). Softmax is a widely used function for vector normalization. It is defined as:

$$softmax(z_i) = \frac{exp(z_i)}{\Sigma_k^{j=1} exp(z_j)}, 1 \leq i \leq d_{out}$$

3.2.3 Training

The goal of training is to learn the optimal weights that can minimize the distance between the model output, $\hat{y}$ and the expected output, y. A popular loss function is the Mean-Squared-Error between $\hat{y}$, and y:

$$E = \frac{1}{N} \sum_{i=1}^{p} \parallel \hat{y}^i - y^i \parallel^2$$

where N is the number of class labels and E is the Mean-Squared Error. If it is a two-way classification problem, then $N = 2$. For probabilistic classifiers, a commonly used loss function is the negative log likelihood J. It ensures that correct answers are assigned with maximal probability and incorrect ones are assigned with minimal probability. The loss J is defined as follows where V is the total number of observations:

$$J(\theta) = -\sum_{j=1}^{|V|} y_j log\hat{y}$$

With the goal being minimization of loss function, this becomes an optimization problem. A number of optimization methods like gradient descent [87] or Adam [88] are used to find a minimum. This is done by first identifying the direction in which the function's slope is increasing the most steeply and then moving in the opposite direction [89] (See figure 3.3 [90]). In case of Mean-Squared-Error, for the i^th coordinate position, we have the output $y^i = W_{ij}x^i + b$. Now, after differentiating both sides with respect to W_{ij} (the only unknown parameter) using chain rule, we get:

$$\frac{\partial}{\partial W_{ij}}(\parallel \hat{y} - y \parallel {}^2) = -2(\hat{y}_i - y_i)x_j$$

where x_j is the input value in the i^th coordinate position.

This derivative gives us the direction to the maximum, so in order to obtain the minimum point, we have to follow the opposite direction of this gradient. Moreover, we need to make sure that this derivative is as close to 0 as possible in order to obtain the minimum of error. After figuring out which direction to go, we still need to know how far to go. This is done by a parameter called learning rate η which moves the gradient towards the minimum value by determining how far each step should go. However, care must be taken when tuning η because if it is too small, the

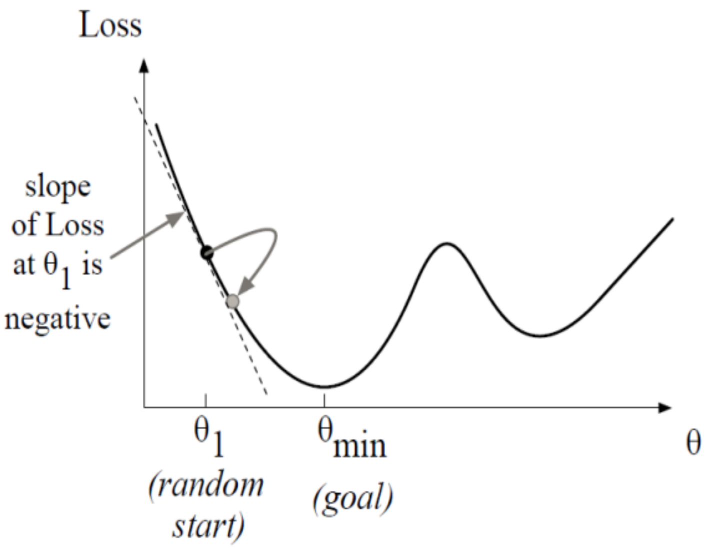


Figure 3.3: θ has been moved in the opposite direction from the slope of the function in order to find the minimum of the loss function [90]

learning will take too long and if it is too large, the weight updates can over-shoot the minimum and not converge. Unfortunately, a learning rate for a certain data set cannot be analytically calculated and can only be known through trial and error. Typical values for a neural network with standardized inputs (inputs mapped to the (0,1) interval) are less than 1 and greater than 10^{-6}. The model's parameters θ are thus updated as follows:

$$\theta^{t+1} = \theta^{(t)} - \eta \nabla_{\theta^{(t)}} J(\theta^{(t)})$$

The backpropagation algorithm [91] uses the chain rule of differentiation to compute the gradient. It takes the partial derivative of the loss function with respect to each parameter in the model. The algorithm first propagates a chunk of the data as input through the network and then calculates the average of the overall loss. After computing the gradients, it updates the weights of the output layer. Next, the error is propagated backwards and the weights of the input and hidden layer is updated. These steps are repeated with the next chunk of training data. If the loss function J is within tolerance, the algorithm terminates. Otherwise, it continues with an another epoch (a complete presentation of the dataset).

3.2.4 Popular ANN Models

Artificial neural networks are widely used to solve various problems in the fields of computer vision, speech recognition, machine translation, or medical diagnosis. It is also very popular in Natural Language Processing (NLP) which deals with the analysis and use of human languages by a machine. NLP helps computers interact with humans by typically reading and generating natural text. With growing interests in this field, new NLP techniques are allowing computers to understand the complexities in grammar, rules, and vocabulary in multiple languages like English, French, German, or Arabic. A number of ANNs are now utilized to solve tasks like Question-Answering [92], Text Summarization [93], Dialog Generation [94], or Text Classification [95]. We will now give a brief summary of some of these neural network models:

1. RNN: In Recurrent Neural Network (RNN), the output from the previous step is fed as input to the current step. In traditional NNs, all the inputs and outputs are independent of each other. This becomes a problem when dealing with sequential data, where one data point depends upon the previous data point. Thus, RNNs came into existence. They have the concept of 'memory' that helps them store the states or information of previous inputs to generate the next output of the sequence. The landscape of chatbots has evolved a lot as a result of this simple technique [96]. RNNs can help chatbots comprehend the conversational environment for interpreting user's inputs and delivering contextually correct responses.

2. Seq2Seq Neural Models: The Sequence to Sequence (Seq2Seq) model consists of two RNNs, an encoder that processes the input, and a decoder that creates the output [97]. It is widely used in the industry for response generation [98]. Once

the model is fed a variety of input sentences, the encoder encrypts the input text and the decoder decrypts it to produce the desired output. The model is most commonly employed in language translation, where the input and output sentences are in two different languages. This approach can also be used to convert between the inputs and outputs in chatbots [98].

3. LSTM: Long short-term memory networks (LSTM) are a special kind of RNN [99] capable of handling long-term dependencies. It tackles the vanishing gradient problem of a typical RNN with the help of memory cells and gates. These two components allow LSTMs to recall past information for extended periods of time. Memory cells are like the memory of a computer that can store, write, and read information. The gates consist of input gates, forget gates, and output gates used to control the flow of information. Even when there is a long period of gap between major events, a well-trained LSTM network can perform outstanding categorization, processing, and prediction of time series. Because LSTMs can regularly refer to a piece of distant information in time, they are extremely valuable in creating chatbots [99, 100].

4. Transformer: It is a deep learning model that adopts self-attention mechanism to differentially weight the significance of each part of the input data. Like RNNs, transformers can process sequential input data. Unlike RNNs, transformers process the entire input all at once instead of processing one word at a time. This allows for more parallelization and reduces training time [101]. The attention mechanism provides context for any position in the input sequence. The additional training parallelization allows training on larger datasets. This led to the development of pretrained systems like BERT and GPT, which were trained on large language datasets and can be fine-tuned for specific tasks. Thus, transformer-based models are achieving SOTA scores on many NLP tasks like Dialog Generation [102], Question Answering [103], Translation [104] and are

quickly replacing other RNN based models like LSTMs.

It is to be noted that both of our chatbots, ANA and MIRA, use transformer-based models to achieve superior performance.

3.3 Popular Classification Algorithms

In the previous section, we only talked about neural networks. In this section, we discuss some of the popular classification algorithms that are often used as baselines for a number of NLP tasks.

1. Logistic Regression (LR): It estimates the probability of the occurrence of an event based on a given dataset of independent variables. It is used to predict a binary outcome (e.g., whether something happens or does not). In LR, the independent variables can be categorical or numeric, but the dependent variable is always categorical. Since the outcome is a probability, the dependent variable is bounded between 0 and 1. Mathematically,

   $$P(Y = 1|X) \ or \ P(Y = 0|X)$$

   Formally, it calculates the probability of dependent variable Y, given independent variable X.

2. Naive Bayes: This classification algorithm is based on the Bayes theorem:

   $$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

   Using this theorem, we can find the probability of an event A happening, given that B has occurred. Here, B is the evidence and A is the hypothesis. Naive Bayes assumes that all the features are independent and the presence of one does not affect the other. However, this is not true in most of the events in real life. Hence, it is called naive.

3. K-Nearest Neighbors (KNN): It is a simple, supervised ML algorithm used for classification, regression, and missing value imputation. It is based on the idea that the observations closest to a given data point are the most ‘similar’ observations in a data set and we can therefore classify unforeseen points based on the values of the closest existing points. K is the number of nearest neighbors. By choosing K, the user selects the number of nearby observations to use in the algorithm. For classification, a majority vote is used to determined which class a new observation should fall into. Larger values of K are often more robust to outliers and produce more stable decision boundaries than very small values.

4. Decision Tree: It is a supervised learning algorithm that is used for classification. It works like a flow chart, separating data points into two similar categories at a time from the ‘tree trunk’ (root node) to ‘branches’ (decision nodes) to ‘leaves’ (leaf nodes) where the categories become more finitely similar. This creation of categories within categories, allows for organic classification with limited human supervision. Random forest algorithm is an expansion of decision tree. It involves constructing a multitude of decision trees at first with the training data, then fitting the new data within one of the trees as a ‘random forest’. It essentially averages the data and then connects it to the nearest tree on the data scale. Unlike decision tree, Random forest models do not ‘force’ data points within a category unnecessarily.

5. Support Vector Machines (SVM): SVM uses algorithms to train and classify data within degrees of polarity. Figure 3.4 shows two tags red and blue with two data features X and Y. The classifier is trained to output a new X/Y coordinate as either red or blue. Here, SVM assigns a line as a hyperplane to best separate the tags. However, as datasets become more complex, a simple line may not be enough (see Figure 3.5). Using SVM, we can classify complex data by converting them into linearly separable data in higher dimensions (3D)

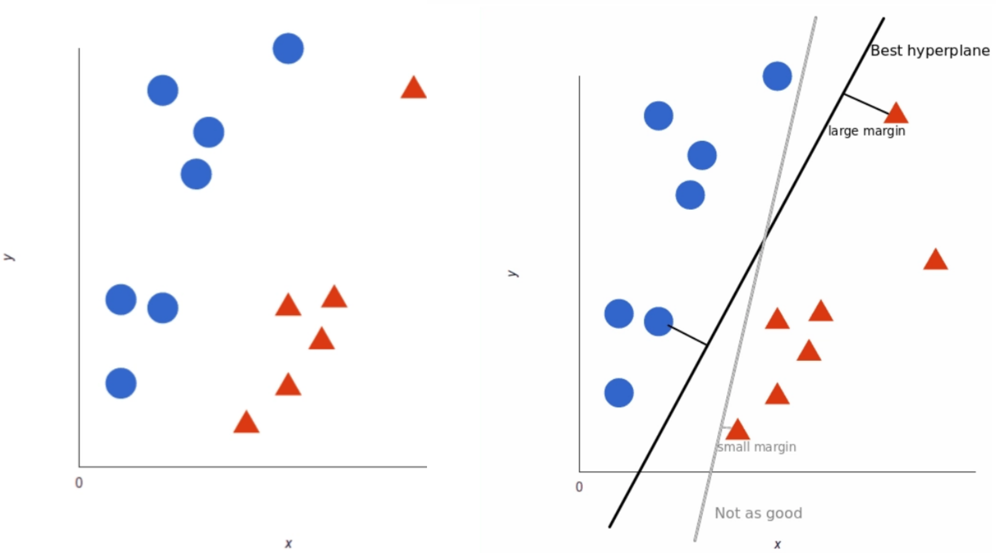


Figure 3.4: SVM classifier on 2D plane [105]

and later project the decision boundary back to the original dimensions (2D). We will be discussing SVM in more details later.

3.4 Classifiers Used

For our chatbots, we use two types of classifiers to detect the dialog acts of our users. In AI and ML, classification refers to the machine's ability to assign labels to their corresponding examples. A classifier is able to decide how to assign an instance to its group by learning the patterns of that assignment from the training features available in a labeled training data set. In NLP, text classification refers to the process of labeling or organizing text data into groups. Dialog act recognition is a type of text classification task. Classification can be of two types:

1. Binary Classification: Here, the machine should classify an instance as only one of two classes: yes or no, 1 or 0, true or false etc. For example: detecting whether an email is spam or not.

2. Multi-class Classification: Here, the machine should classify an instance as only

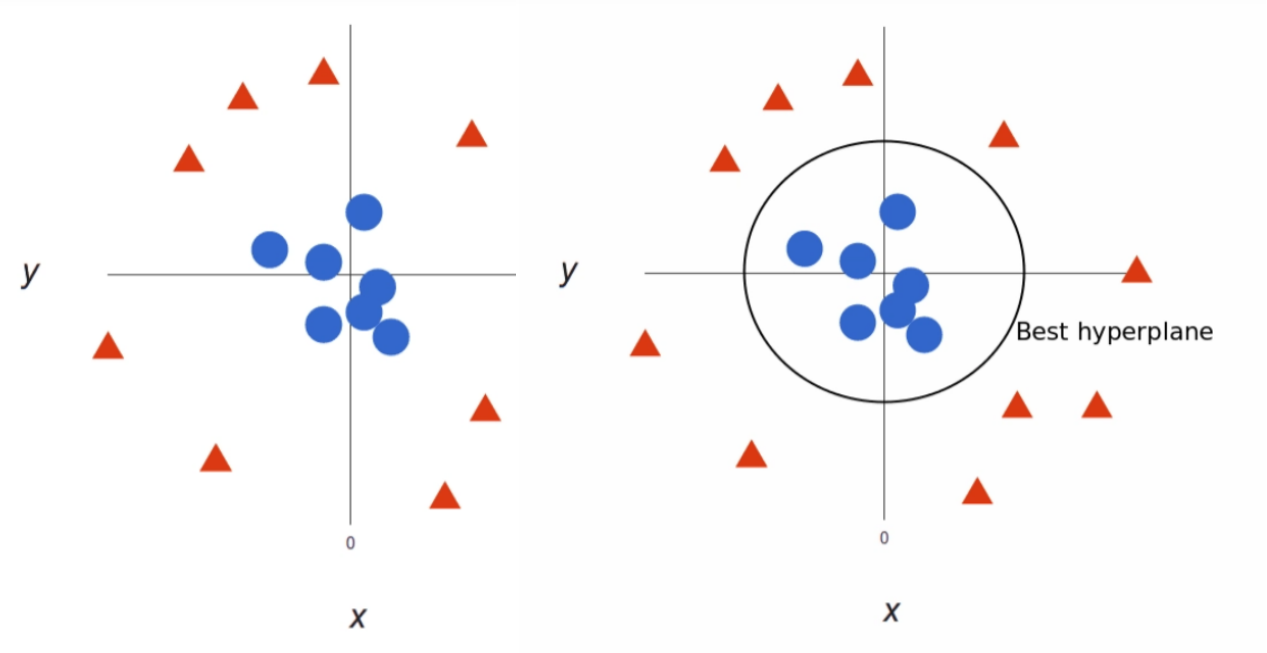


Figure 3.5: SVM classifier on 3D plane mapped into 2D [105]

one of three or more classes. For example: classifying a tweet as positive, negative, or neutral.

From the above definitions, it is clear that our dialog act detection task is a multi-class classification problem, meaning, we need to accurately label a given user input as one of the 8 identified dialog acts: yes/no question, factual question, direct order, indirect order, statement, feedback, apology, greeting. Based on the related works discussed previously, we decided to implement two of the most widely used NLP models for DAC classification: SVM and BERT.

3.4.1 SVM

As briefly discussed earlier, Support Vector Machine (SVM) is a supervised learning model with associated learning algorithms used for data classification and regression. It was developed by Vladimir Vapnik at ATT Bell Laboratories with a number of colleagues (Boser et al. [106], Guyon et al. [107], Cortes and Vapnik [108], Vapnik et al [109]). The objective of SVM is to find a hyperplane in an N-dimensional space (where N is the total number of features) that distinctly classifies the data points. In other words, it aims to optimize the width of the gap (i.e. the maximum margin hyperplane) between classes. This provides some reassurance for future data points to be classified with more confidence. The dimension of the hyperplane depends upon the number of features. If the number of input features is 2, then the hyperplane is a line. If the number of input features is 3, then the hyperplane becomes a plane and so on. The data points closer to the hyperplane are called support vectors. They influence the position and orientation of the hyperplane. Using support vectors, we maximize the margin of the classifier.

SVMs are universal learners [110] and one of the most robust prediction methods. In their basic form, SVMs learn a linear threshold function. However, by a simple ‘plug-in’ of an appropriate kernel function (a function that determines the smoothness and efficiency of class separation), they can be used to learn polynomial classifiers, radial basic function (RBF) networks and three-layer sigmoid neural nets. Moreover, SVM’s ability to learn is independent of the dimensionality of the feature space. They measure the complexity of hypotheses not based on the number of features but based on the margin with which they separate the data. Because of this, if our data is separable with a wide margin using functions from the hypothesis space, we can generalize even in the presence of many features. As Joachims [110] points out, SVM is perfectly suitable for text categorization as they are able to efficiently handle high dimensional input space, few irrelevant features and sparse document vectors. Moreover, since text categorization problems are mostly linearly separable, SVM is the perfect candidate.

Although natively SVM does not support multi-class classification, the same principle can be utilized after breaking down the multi-classification problem into multiple binary classification problems. There are two approaches to do so: in One-to-One approach, data points are mapped to high dimensional space to gain mutual linear separation between every two class. In other words, a binary classifier is used per each pair of classes. Here, the classifier can use m SVMs and each SVM would predict membership in one of the m classes. In the One-to-Rest approach, the breakdown is set to a binary classifier per each class and the the classifier can use m(m-1)/2 SVMs.

3.4.2 BERT

Bidirectional Encoder Representations from Transformers (BERT) is an ML model developed in 2018 by researchers at Google AI Language [111]. It serves as a reliable solution to the most common NLP tasks like sentiment analysis and named entity recognition. It is composed of several transformer encoders stacked together. Transformers utilize an attention mechanism that learns contextual relations between words (or sub-words) in a text. In its vanilla form, Transformer includes two separate mechanisms- an encoder that reads the text input and a decoder that produces a prediction for the task. Since BERT's goal is to generate a language model, only the encoder mechanism is necessary. Further, each Transformer encoder in BERT is composed of two sub-layers: a feed-forward layer and a self-attention layer. Unlike directional models (reads input sequentially from left-to-right or right-to-left), BERT is bidirectional. That is, the Transformer encoder reads the entire sequence of words at once. This allows the model to learn the context of a word based on all of its surroundings (left and right of the word), just like humans do.

As opposed to taking a less effective directional approach (predicting the next word in a sentence) to training, BERT uses two unique training strategies: Masking and Next Sentence Prediction (NSP). In the Masking approach, before feeding word sequences into BERT, a certain percentage of the words in each sequence are replaced with a [MASK] token. The model then attempts to predict the original value of the masked words, based on the context provided by the other, non-masked, words in the sequence. The BERT loss function takes into consideration only the prediction of the masked values and ignores the prediction of the non-masked words. As a consequence, the model converges slower than directional models, a characteristic which is offset by its increased context awareness. In the second training approach, NSP, the model receives pairs of sentences as input and learns to predict whether the second sentence in the pair follows the first sentence in the original document. During training, 50% of the inputs are a pair in which the second sentence is the subsequent sentence in the original document, while in the other 50% a random sentence from the corpus is chosen as the second sentence. It is assumed that the random sentence will be disconnected from the first sentence. When training the BERT model, Masked LM and NSP are trained together, with the goal of minimizing the combined loss function of the two strategies.

Google's BERT has also made headlines for famously introducing the pre-training/fine-tuning paradigm: after pre-training in an unsupervised manner on massive corpus, the model can be quickly fine-tuned on a specific downstream task with relatively fewer labels, because it has already learnt the general linguistic patterns during pre-training. Devlin et al. [112] in the original paper proposed fine-tuning a pretrained BERT by simply adding an additional layer after the final BERT layer and training the network for a few epochs. With this technique, the authors demonstrated strong performance on standard NLP benchmark problems like GLUE, SQuAD, and SWAG, after fine-tuning for just 2-3 epochs with the ADAM optimizer, with learning rates between 1e-5 to 5e-5. Because of its remarkable success, this pre-training/fine-tuning paradigm has become a standard practice in NLP. However, fine-tuning large pre-trained language models however is not a new concept. Although language models usually work well on generic text, often times they do not fit well when used in a specific domain. For example, using it in a medical or scientific domain which has its peculiar language. For this purpose, domain adaptation of model is required, which essentially means training a pre-trained model on a new task specific dataset in order to obtain more accurate predictions. As a result of the fine-tuning procedure, the weights of the original model are updated to account for the characteristics of the domain data. It is an incredibly powerful technique that reduces computation costs, carbon footprint, and enables the use of state-of-the-art models without having to train one from scratch.

3.5 Performance Metrics Used

This section provides a brief overview of the metrics we use to measure the performance of our dialog act classifiers.

1. Accuracy: Accuracy is the fraction of correct predictions of our model. It is one of the most popular metrics to evaluate classification models. It describes how the model performs across all classes and is useful for balanced classification tasks. Mathematically,

   $$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

   For binary classification, accuracy can also be calculated in terms of positives and negatives as follows:

   $$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

   Where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

2. Confusion Matrix: In ML, a confusion matrix or error matrix, is a specific table layout that is typically used to visualize the performance of a supervised learning algorithm. The matrix rows represent the instances in an actual class while the matrix columns represent the instances in a predicted class or vice versa. The name refers to the fact that it is easier to identify whether the system is confusing between two classes (i.e., commonly mislabeling one as another). It uses 4 important terms:

   (a) True Positives: These are cases in which we predicted YES and the actual output was also YES.

(b) True Negatives: These are cases in which we predicted NO and the actual output was also NO.

(c) False Positives: These are cases in which we predicted YES but the actual output was NO.

(d) False Negatives: These are cases in which we predicted NO but the actual output was YES.

3. F1 Score: F-score or F-measure assesses the predictive skill of a model by elaborating on its class-wise performance rather than its overall performance (accuracy). It is the harmonic mean of precision and recall. Precision is the number of true positive results divided by the number of all positive results, whereas recall is the number of true positive results divided by the number of all instances that should have been identified as positive. Mathematically,

$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

$$F1 = \frac{2TP}{2TP + FP + FN}$$

The highest possible value of an F-score is 1.0, indicating perfect precision and recall, and the lowest possible value is 0, if either precision or recall is zero.

4. Standard Deviation: In statistics, the standard deviation is a measure of the amount of variation or dispersion of a set of values from the mean value. A low standard deviation indicates that the values are close to the mean of the set, while a high standard deviation indicates that the values are spread out over a wider range. Mathematically,

$$s = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2}$$

where x_i = value of the ith point in the data set, $\overline{x}$ = the mean value of the data set, and N = number of data points in the dataset.

During cross-validation or bootstrapping, it is a good idea to measure the standard deviation of the accuracy in order to understand the consistency of the model on different data samples.

5. Macro Average: It is concerned with aggregations or totals and gives equal weight to each category. This results in a bigger penalisation when a model does not perform well on minority classes. It is especially useful for datasets with class imbalance. We report the macro average precision, recall and F1-score for each of our classes in all of our experiments using the following formulae:

$$Recall_{MacroAvg} = \frac{(Recall_1 + Recall_2 + ... + Recall_n)}{n}$$

$$Precision_{MacroAvg} = \frac{(Prec_1 + Prec_2 + ... + Prec_n)}{n}$$

$$F1_{MacroAvg} = \frac{F1_1 + F1_2 + ... + F1_n}{n}$$

Chapter 4

Experimentation

In this chapter, we discuss in depth the architecture of our DAC classifiers and report the results obtained through extensive experiments. We further analyze the results and draw conclusions.

4.1 Experimental Setup

As mentioned earlier, we use two dialog act classifiers for our experiments. The first one is the SVM model which is our baseline. We used LinearSVC as our classifier. It is similar to SVC with parameter kernel='linear', but provides more flexibility in the choice of penalties and loss functions in Scikit-learn [113]. We used a linear kernel because text classification problems are often linearly separable. Moreover, linear kernel is good when there is a lot of features [110] and as we know, text classification has a large numbers of instances (document) and features (words). Training an SVM with a linear kernel is also much faster than with any other kernels. Last but not the least, linear kernel in comparison has fewer parameters to optimize. To convert the text files into numerical feature vectors, we used the Bag-of-Words technique (CountVectorizer) and later ran the TF-IDF technique (TfidfTransformer) over the features generated by Bag-of-Words. Next, we trained our classifier and ran it on the test dataset to measure the rate of accuracy.

Our second DAC classifier is based on the pretrained BERT model [112]. We chose

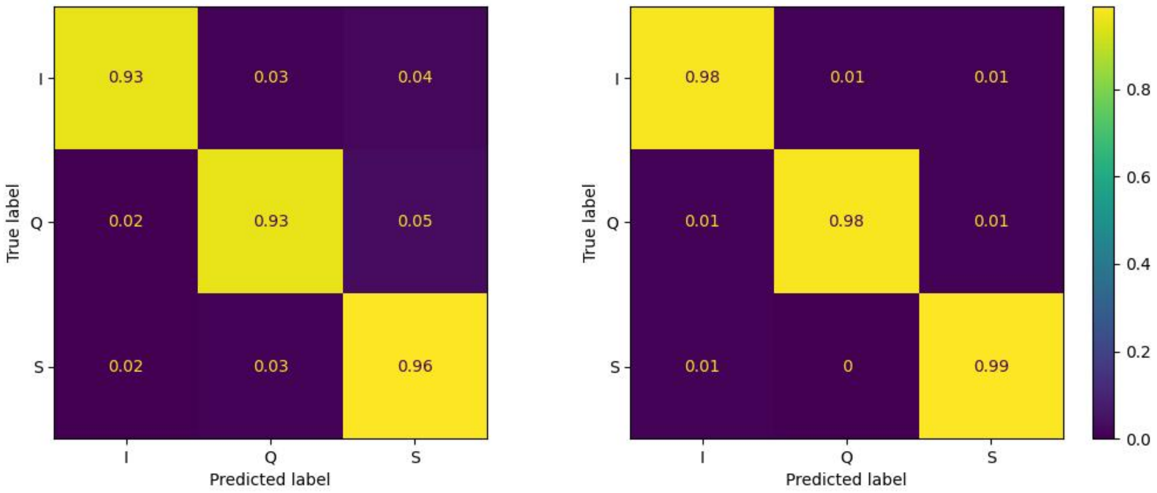
Model	Dataset	Accuracy	Precision	Recall	F1-Score
SVM	3 DA	0.94	0.94	0.94	0.94
BERT	3 DA	0.98	0.98	0.98	0.98

Table 4.1: Performance of our SVM and BERT classifier for detecting the 3 Dialogue Acts

BERT because it is a very good pre-trained language model which helps machines learn excellent representations of text with respect to context in many natural language tasks and thus outperforms the state-of-the-art. Moreover, because BERT was pre-trained on a huge corpus, it can easily be fine-tuned on a new dataset and achieve great results. Before applying BERT, we had to first convert our labels into categorical data. We then loaded the pretrained 'bert-base-cased' model from Tensorflow and later fine-tuned it on the training dataset. We use the corresponding tokenizer with a max length set to 70. The Bert layers accept three input arrays, inputIds, attentionMask and tokenTypeIds. But since tokenTypeIds is necessary only for the question-answering model, we do not pass the tokenTypeIds and only worked with two input layers. We used GlobalMaxPooling1D and then a dense layer to build the CNN layers using hidden states of BERT. These CNN layers yield the output. We used the Adam optimizer with a learning rate of 5e-05, a decay of 0.01 and 'CategoricalCrossentropy' as loss since we passed the categorical labels as the target. Once training is complete, we calculate the accuracy on the test dataset.

4.2 Experiments on 3 DACs

Our initial dataset was massive and comprised of 3 dialog acts, statement, question, and imperative as proposed by Quinn et al.[30]. Upon training our classifiers on our training data, we tested their performance on our test data. From Table 4.1, we can see that our baseline SVM classifier yielded an accuracy of 94%. Moreover, our BERT-based model outperformed the baseline by achieving an accuracy of 98%. For further analysis, we take a look at the confusion matrices in Figure 4.1 and notice that, in case



(a) SVM

(b) BERT

Figure 4.1: Confusion matrices of our two classifiers for detecting the 3 Dialogue Acts

of SVM, the accuracy for all 3 classes is above 90% with the highest being Statement (96%). The model, however, tends to misclassify around 4% of Imperatives (I) and 5% of Questions (Q) as Statements (S). Upon analyzing the misclassified sentences, we found instances where sentences like ‘So, uh, don’t operate heavy machinery.’, ‘Do people regret their tattoos often even if it’s a good tattoo?’ would get mislabelled as Statements. As a result of that, the Imperative and the Question class have an accuracy of 93%. Our BERT-based model, on the other hand, achieves 98% accuracy on the Imperative and the Question class and an impressive 99% accuracy for the Statement class. However, the model too misclassifies a few sentences like ‘Oh, how little!’, and ‘Why don’t you state it?’ due to their unconventional structure.

4.2.1 Generalizability of Model

The generalizability (or robustness) of a model is a measure of its successful application to data sets other than the one used for training and testing. To evaluate the generalizability of our dialog act classifiers, we decided to test our trained models on Quinn et al.’s proposed dataset. This is because, although we both share the same dialog acts, our datasets were curated from completely different sources. Table 4.2 gives a brief summary of the results were obtained. For a fair comparison of performance, we use our proposed classifiers in all the experiments regardless of the dataset in question.

We start off with our baseline SVM classifier which Quinn et al. had also used in their work. When the model was trained and later tested on their dataset, we achieved an accuracy of 72% instead of the reported 82%. This might be due of some differences in the environmental setup. And since their paper does not provide much detail on this, we could not replicate it. As previously mentioned, our baseline model achieves an accuracy of 94% when trained and tested on our dataset. Upon training the baseline model on our dataset, we run it on Quinn et al.'s test dataset and observe an accuracy of 82%. Although the drop is almost 12%, from the confusion matrix in Figure 4.2, we can see that accuracy for classes Question and Statement is still above 70% (74%). Moreover, the Imperative class has the highest accuracy rate of 93%. On the flip side, upon training the SVM classifier using Quinn et al.'s train dataset and later testing it on our test dataset, the accuracy of the model drops down to 66%. The model especially struggles when it comes to identifying Imperatives (49%) and Questions (60%). From these observations, we can say that, the SVM classifier does not generalize that well on unseen data and is highly dependent on the data it is trained on. Despite that, the experiment shows that our curated dataset is so large and diverse that upon using it to train the baseline model, an accuracy above 80% is observed on unseen data. The same, however, cannot be said about the dataset and the corresponding model proposed by Quinn et al.

Next, we repeat the same set of experiments with our BERT-based classifier. We first fine-tune it with Quinn et al.'s train data and later run it on their test data. An accuracy of 89% is obtained which is much better than the baseline (a 17% jump from 72%). Repeating the same process with our massive training and test data yields a high accuracy of 98%. Now comes the more interesting observations where we compare

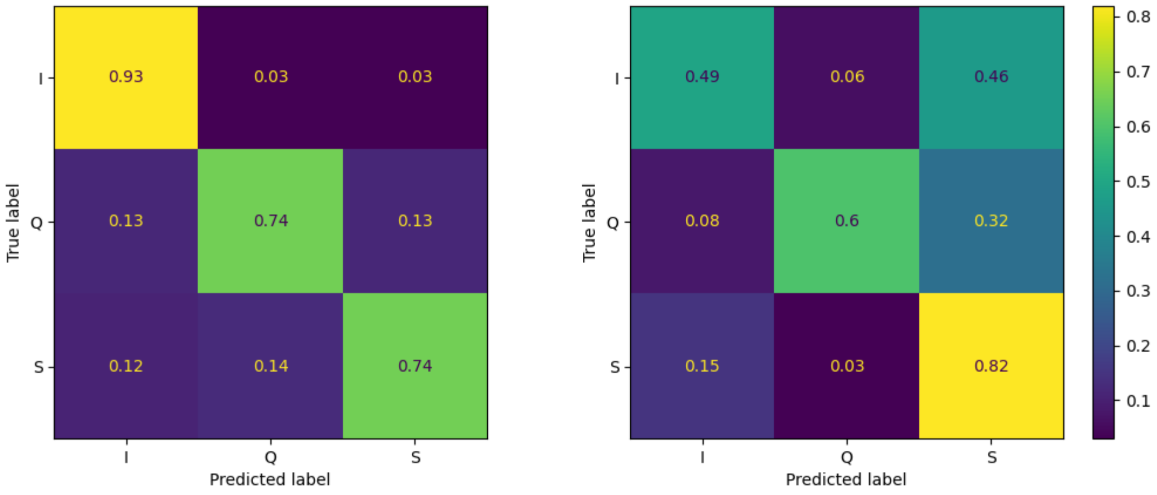
Model	Train Dataset	Test Dataset	Accuracy	Precision	Recall	F1-Score
SVM	Quinn Train	Quinn Test	0.72	0.74	0.71	0.71
	Our Train	Our Test	0.94	0.94	0.94	0.94
	Our Train	Quinn Test	0.82	0.82	0.82	0.82
	Quinn Train	Our Test	0.66	0.69	0.63	0.65
BERT	Quinn Train	Quinn Test	0.89	0.90	0.88	0.89
	Our Train	Our Test	0.98	0.98	0.98	0.98
	Our Train	Quinn Test	0.90	0.91	0.89	0.89
	Quinn Train	Our Test	0.86	0.90	0.83	0.84

Table 4.2: Comparing the performance of our classifiers for detecting 3 DACs upon using Quinn et al.'s dataset

the generalizability of our BERT-base DAC classifier. When the model trained on our dataset is run on Quinn et al.'s test dataset, we observe an accuracy rate of 90%. Although it drops by 8%, nonetheless, an accuracy of 90% on an unseen dataset curated from completely different data sources is still very impressive. Moreover, from Figure 4.3, we can observe that the individual accuracy per class is also very high (Question: 98%, Imperative: 90%, and Statement: 79%). On the flip side, when the model is trained on Quinn et al.'s dataset and then evaluated on our test dataset, we observe an accuracy of 86% which does not seem that bad at first. However, if we take a look at the confusion matrix, we can see that for the Imperative class, the individual accuracy is only 56% which is not acceptable. This further proves the generalizability of our BERT-based DAC classifier and validates the high-quality and diverse nature of our training data. The experiments further suggest that, overall, BERT-based models are far more generalizable than SVM models for DAC classification on unseen data. Unlike SVM, BERT is not completely dependent on the training dataset.

4.3 Experiments on 8 DACs

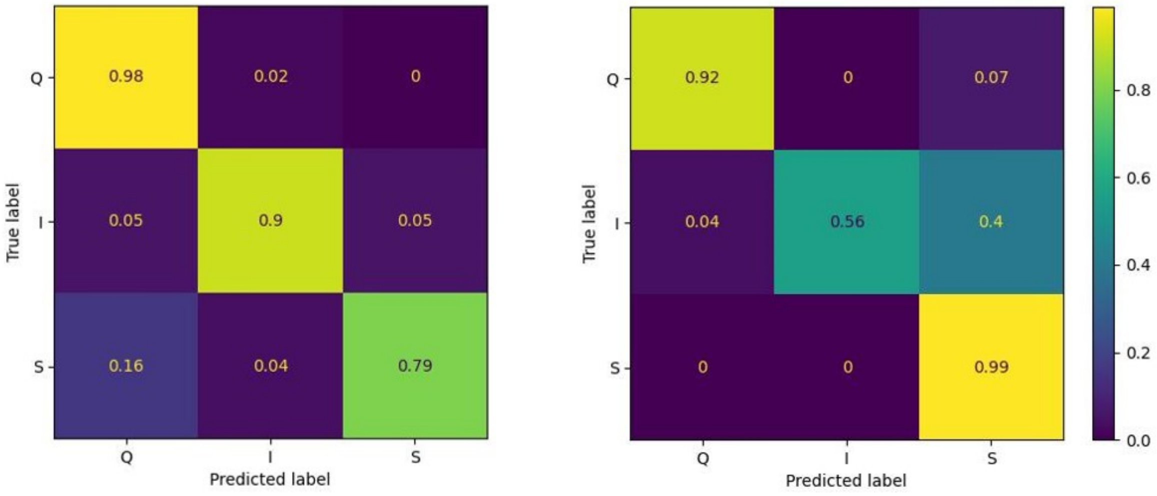
We will now evaluate the performance of our classifiers on our proposed dataset with 8 dialogue acts. From Table 4.3, it is evident that both our baseline SVM and classifier BERT perform extremely well on the dataset, SVM yielding an accuracy of 96% and BERT outperforming SVM by 3%, achieving an accuracy of 99%. One of the main



(a) Our Train Data, Quinn Test Data

(b) Quinn Train Data, Our Test Data

Figure 4.2: Confusion matrices of our SVM classifier for identifying 3 DACs on the two datasets



(a) Our Train Data, Quinn Test Data

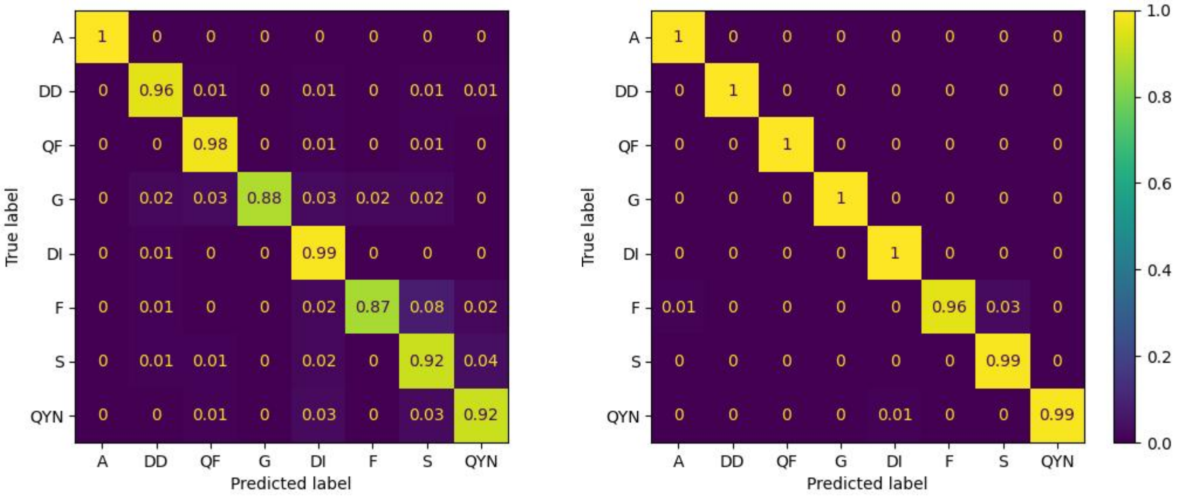
(b) Quinn Train Data, Our Test Data

Figure 4.3: Confusion matrices of our BERT-base classifier for identifying 3 DACs on the two datasets

reasons for such a high accuracy rate might be because of the stark differences in the structure of sentences for each of the labels. For further analysis of the wrongly labelled data, we take a look at the confusion matrices in Figure 4.4. The y-axis shows the true labels of the examples, whereas the x-axis shows the predicted labels.

At first, we take a look at the majority classes that have an average of approximately 3771 examples per label. For classes like Directive Direct Order (DD), Question Factual (QF), and Directive Indirect Order (DI), SVM achieves individual accuracies of 96%, 98%, and 99%, with only a very few instances of misclassification. For example, 0.2% of Greetings (G) are misclassified as Direct Orders. However, the accuracy is comparatively low for other majority classes like Statement (S) and Question Yes/No (QYN) (92%). Upon further analysis, we see that 8% of Feedbacks (F) are being misclassified as statements. For example, 'This works well ', 'I'm glad you are my friend' are all Feedbacks but are misclassified as Statements. This makes sense given the similarity in sentence structure for both of these classes. In case of Yes/No questions, however, the low accuracy rate comes from misclassifying a lot of the Statements (4%) and Feedbacks (2%). For example, 'My issue is that there is always drama' and 'It is good' has been misclassified as Yes/No Questions. One of the reasons for this might be the presence of the helping verb 'is' in the beginning of the sentence, which often represents the structure for a Yes/No Question ('Is it cold in here?'). The baseline model fails to learn the difference in some cases.

Now we take a look at the minority classes. Although the class Apology (A) has a very small number of train examples (73), SVM is able to detect all of them accurately. This might be because of how little the examples of this class vary from each other. For classes like Greetings and Feedback, however, the accuracy is pretty low in comparison, 88% and 87%. Upon further analysis, we noticed that a number of examples labelled as Greetings are misclassified as Direct Orders ('See ya!'), Factual Questions ('What is up, Mira?'), Indirect Orders ('Top of the morning to you!'), Feedback ('It was nice meeting you!') and even Statements ('Sira, how's it go-



True label	A	DD	QF	G	DI	F	S	QYN
A	1	0	0	0	0	0	0	0
DD	0	0.96	0.01	0	0.01	0	0.01	0.01
QF	0	0	0.98	0	0.01	0	0.01	0
G	0	0.02	0.03	0.88	0.03	0.02	0.02	0
DI	0	0.01	0	0	0.99	0	0	0
F	0	0.01	0	0	0.02	0.87	0.08	0.02
S	0	0.01	0.01	0	0.02	0	0.92	0.04
QYN	0	0	0.01	0	0.03	0	0.03	0.92

(a) SVM

True label	A	DD	QF	G	DI	F	S	QYN
A	1	0	0	0	0	0	0	0
DD	0	1	0	0	0	0	0	0
QF	0	0	1	0	0	0	0	0
G	0	0	0	1	0	0	0	0
DI	0	0	0	0	1	0	0	0
F	0.01	0	0	0	0	0.96	0.03	0
S	0	0	0	0	0	0	0.99	0
QYN	0	0	0	0	0.01	0	0	0.99

(b) BERT

Figure 4.4: Confusion matrices of the two classifiers for identifying 8 DACs

ing?'). Similar is the case for Feedback. Despite having more training examples than Apology, they are not enough for the baseline model to learn the specific patterns to recognize them. Two mainly reasons for this might be because of the high variations in user feedback as well as their similarity in structure with sentences belonging to the Statement class. Future work might look into using rules to detect these minority classes instead and compare the performance.

Lastly, we take a look at our fine-tuned pretrained Bert-base model. Unlike SVM, BERT does a very good job achieving 99% individual accuracy for almost all of the classes (Apology, Direct Orders, Factual Questions, Greetings, Indirect Orders, Feedback, Statement, and Yes/No Question). A slight drop (96%) in performance is observed for Feedback class- most of which are often misclassified as Statements just like SVM.

4.3.1 Model Generalizability

One of the biggest issues with building a DAC classifier is making sure that it not only performs well on the proposed dataset but also generalizes well onto completely

Model	Dataset	Accuracy	Precision	Recall	F1-Score
SVM	8 DA	0.96	0.96	0.94	0.95
BERT	8 DA	0.99	0.99	0.99	0.99

Table 4.3: Performance of our SVM and BERT classifier for detecting the proposed 8 Dialogue Acts

unseen datasets. However, as mentioned earlier, researchers propose different DAC tag-sets based on what they think are suitable for their particular task. As a result, evaluating the generalizability of a specific DAC classifier becomes incredibly difficult. This is because, it requires finding a new dataset that is completely different from the training and test data but also shares the same set of highly specific DACs. To tackle this, we first selected a pre-existing dataset that was never used for curating our train or test data. Next, we manually labelled it with our proposed taxonomy of 8 dialog acts. We will be referring to this dataset as 'generalized dataset'.

We chose the DialogueSum dataset proposed by Chen et al. [114] as the source for our generalized dataset. It is a large-scale dialogue summarization dataset, consisting of 13,460 dialogues from three public dialogue corpora, namely Dailydialog [45], DREAM [115] and MuTual [116], as well as an English speaking practice website. The dataset contains face-to-face high quality spoken dialogues from a wide range of daily-life topics including schooling, work, medication, shopping, leisure, travel and so on. Most of the conversations take place between friends, colleagues, and between service providers and customers. This unfortunately is a downside for us because we trained our model on a dataset that has conversations that a user would have with a chatbot, not a person. We mitigated this issue by deciding to only include exchanges that a user is more likely to have with a chatbot. For example: sentences like 'Zach, what's that on your arm?', 'Here, let me help you with your coat and we'll be on our way.' were avoided. Moreover, given how large the dataset is, we only chose a few samples for each dialog act manually. Fortunately, the dataset had a good number of examples per class to choose from. This in a way shows that the eight dialogue acts that we identified are not only appropriate for our chatbots but are also applicable for a wide range of domains and tasks. DialogueSum, however, had more examples for some class than other- with Statement class being the most dominant one. On the other hand, examples for the class Direct Order were a bit difficult to find. This is because people do not tend to give orders to each other during casual conversations and even when they do, the type of order they give to a person is different from the type of order they give to a chatbot. Regardless, we were able to find examples that can be considered as orders a user might give to a chatbot. In the end, we curated a generalized dataset with 8 Apologies, 9 Greetings, 9 Feedback,30 Indirect Orders, 36 Direct Orders, 43 Factual Questions, 45 Yes/No Questions and 47 Statements.

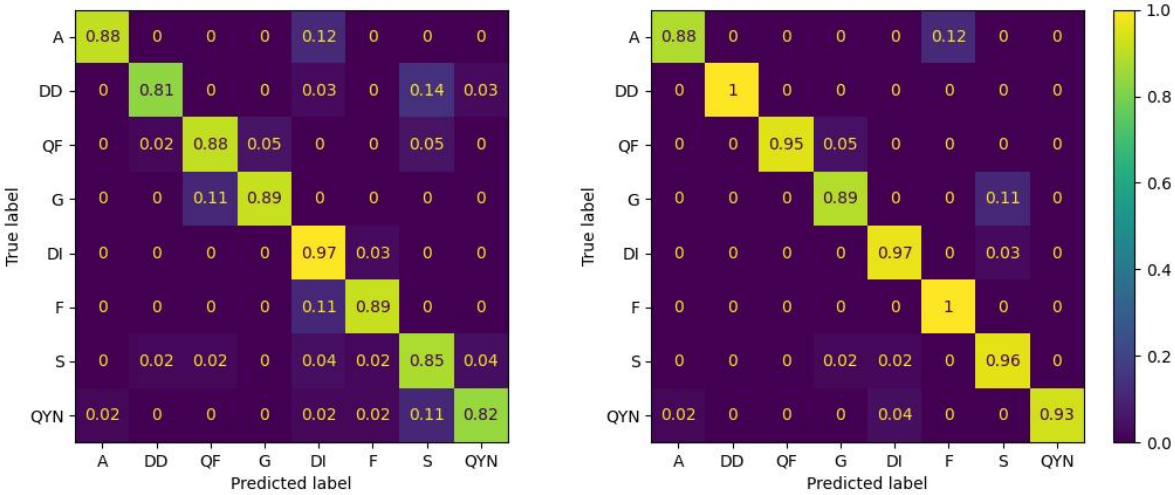
Like before, we first train our baseline SVM model and fine-tune our pretrained BERT-based classifier on our proposed train data. Next, we evaluate their performance by running them on the generalized dataset. Table 4.4 shows the results of the experiment. It is seen that the performance of both the models drop when tested on the generalized dataset. As mentioned earlier, given the difference in the two datasets, this is quite expected. Despite the drop in overall accuracy, both the models still perform very well. Although our baseline SVM classifier has an accuracy of 86% (a drop of 10 points), our BERT-based model still retains an impressive accuracy of 96% (a mere drop of 3 points). This proves that both of our models, especially our fine-tuned BERT-based model, are more or less generalizable and robust on unseen data.

For further analysis, we take a look at the examples that were mislabelled. Figure 4.5 shows the confusion matrix for both the models. In case of SVM, we notice that the model struggles the most when it comes to identifying Direct Orders. For example: sentences like ‘Please wrap it for me and I’ll take it’, ‘Go back to sleep then but only five more minutes’, ‘Just turn down the TV set a little so that it wont be so noisy’ etc. are mislabelled as either Statements or Yes/No Questions. A possible reason for this might be because most of the Direct Orders this model was trained on were straightforward and comprised of some very common commands that chatbots are generally tasked with, e.g playing a song, booking a flight, reserving a seat etc. As a result, the model has a hard time associating the newer examples as orders. The baseline model also struggles with classifying a number of Yes/No questions correctly. For example, sentences like 'Excuse me, do you speak English?', 'Have you turned on the air-conditioner?', 'Does she have a job?' are mislabelled as Statements or Indirect Orders. A possible reason for this might be because the training dataset mostly includes Yes/No questions that are usually factual and not casual (i.e. sentences like 'Is Canada in the United States of America?' instead of 'Do you like to play the piano?'). Moreover, the phrase 'excuse me' in our dataset is associated with the class Apology a number of times which might be the reason for this misclassification. On the flip side, the accuracy rate for the class Indirect Order is very high (97%)- pertaining to the fact that most of the requests made in the DialogSum dataset are similar to the ones one might make to a chatbot e.g: 'Please contact Betty Sue', 'Tell me the fact please' etc. Overall, the accuracy rate for all the classes are above 80%, which is very reasonable.

Now, moving on to our BERT-based DA classifier, we notice the least individual accuracy in the minority classes Apology (88%) and Greeting (89%) for mislabelling two sentences 'I hope you can forgive me' and 'Hi, my name is Susan' as Statement and Indirect Order. Most probable reason for this is the lack of enough training dataset for these two classes. As a result, the model is not able to learn all sorts of variations properly. On the upside, the remaining classes all have an impressive accuracy rate of over 90%, the best performing class like last time is Direct Orders for the same reasons. Despite having an accuracy of 93%, the Yes/No Question class struggles a bit with sentences like 'Have you turned on the air-conditioner', 'Can I exchange it?' etc. since they resemble the structure of Indirect Orders in the training examples to an extent ('Turn on the air-conditioner', 'Exchange it'). All in all, both of our proposed models seem to generalize well on the new dataset despite the slight drop

Model	Dataset	Accuracy	Precision	Recall	F1-Score
SVM	Test	0.96	0.96	0.94	0.95
	Generalized	0.86	0.85	0.87	0.86
BERT	Test	0.99	0.99	0.99	0.99
	Generalized	0.96	0.92	0.95	0.93

Table 4.4: Performance of our SVM and BERT classifier for detecting the proposed Dialogue Acts



	A	DD	QF	G	DI	F	S	QYN
A	0.88	0	0	0	0.12	0	0	0
DD	0	0.81	0	0	0.03	0	0.14	0.03
QF	0	0.02	0.88	0.05	0	0	0.05	0
G	0	0	0.11	0.89	0	0	0	0
DI	0	0	0	0	0.97	0.03	0	0
F	0	0	0	0	0.11	0.89	0	0
S	0	0.02	0.02	0	0.04	0.02	0.85	0.04
QYN	0.02	0	0	0	0.02	0.02	0.11	0.82

(a) SVM

	A	DD	QF	G	DI	F	S	QYN
A	0.88	0	0	0	0	0.12	0	0
DD	0	1	0	0	0	0	0	0
QF	0	0	0.95	0.05	0	0	0	0
G	0	0	0	0.89	0	0	0.11	0
DI	0	0	0	0	0.97	0	0.03	0
F	0	0	0	0	0	1	0	0
S	0	0	0	0.02	0.02	0	0.96	0
QYN	0.02	0	0	0	0.04	0	0	0.93

(b) BERT

Figure 4.5: Confusion matrices of our two classifiers for identifying 8 DACs on the generalized dataset

in accuracy rate. Like last time, we can draw the same conclusion that BERT-based models are more generalizable than SVM models.

Chapter 5

Text Data Augmentation

In the past decade, NLP has achieved tremendous success through the use of neural networks and deep learning models. This progress has often been associated with the rule of more: more data, more complexity, and more computing resource. Training or fine-tuning large dense models for specific domains require substantial amounts of data which is often time consuming, expensive, and difficult to obtain. Researchers have successfully explored a number of solutions to address this issue, data augmentation (DA). Popularized by Computer Vision, DA refers to strategies for increasing the diversity of training examples without explicitly collecting new data. In other words, such techniques artificially generate new data points by slightly modifying the existing data. The idea here is for the augmented data to act as a regularizer and prevent deep learning models from overfitting [32]. When the training data is augmented with the synthetic data generated using DA, the model resorts to learning abstractions of information which are more likely to generalize. Thus, models trained in this way are expected to be more robust to noise.

Given its significance, DA has been widely applied in Computer Vision where new images are generated by simple transformations like flipping, cropping, rotating, or color jittering [117]. However, unlike images, NL is discreet and comprises complex syntactic and semantic structures. This makes the process of generating new examples with the desired invariances more difficult. Moreover, the newly generated data has to preserve the label of the original data [118]. For example, In sentiment analysis task, if the original sentence is labelled as negative, the newly generated sentence should also bear the same sentiment. Otherwise, the quality of data may drop, causing further deterioration of model performance. Despite these limitations, in the recent years, a number of researchers have successfully proposed different DA techniques for a wide range of NLP tasks like text classification, question-answering, summarization and so on. Wei et al. [36] proposed generating a new sentence from an original sentence in four easy ways: randomly inserting a new word, randomly deleting a word, replacing a word with its synonym, and swapping two random words. Upon training classifiers on this augmented data, significant performance boost on different text classification tasks was observed. Similarly, Kumar et al. [35] showed how transformer based pre-trained models like GPT-2, BERT, and BART can be used for conditional DA and improve model performance. In the recent years, more and more researchers have successfully proposed different DA techniques for common NLP tasks. Given how rapidly this area of research is growing, a number of surveys have also come out to help researchers keep up to date with the existing techniques [119–124].

With the goal of looking into the possibility of using DA techniques to improve the performance of our dialog act classifiers, this chapter is dedicated to text DA techniques in NLP. We start off by providing a comprehensive survey of the existing methods by categorizing them using a novel taxonomy. Next, we conduct a comparative performance study by implementing different DA techniques on a number of NLP tasks and share our findings.

5.1 Brief Overview of DA techniques in NLP

In the following subsections, we provide a brief overview of different text data augmentation techniques using a novel taxonomy that divides them into three broad categories, Easy Data Augmentation (EDA), Paraphrase, and Compositional Generation. Figure 5.1 shows how each of these categories are further broken down into

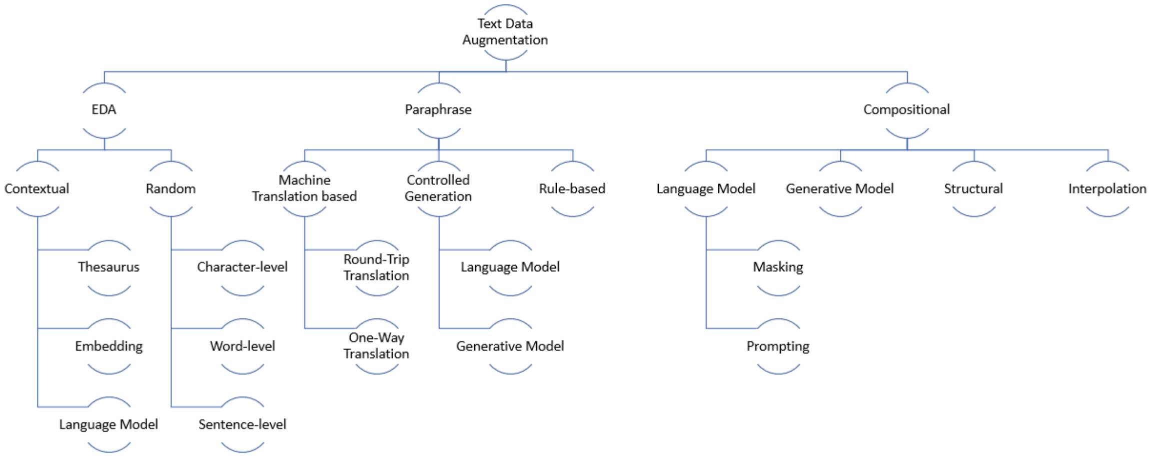


Figure 5.1: Our proposed taxonomy for classifying the different text data augmentation techniques

sub-categories. For a better understanding, Table 5.1 shows examples of the synthetic data generated by applying different augmentation techniques on a single sentence.

5.2 Easy Data Augmentation (EDA)

Famously proposed by Wei et al. [36], Easy Data Augmentation or EDA uses simple but powerful operations for generating synthetic data. Inspired by their work, this category includes DA techniques that are simple and easy to use.

5.2.1 Contextual Replacement

One of the easiest and most popular DA techniques involve generating a new sentence by replacing n number of words with similar words. As the name suggests, this is done by taking the contextual information into account. Here, similar words can refer to synonyms, hyponyms, hypernyms, and words with same Part-Of-Speech (POS) tag.

1. Thesaurus-based: This technique involves finding similar replacement words using a thesaurus derived from WordNet, VerbNet, etc. Kolomiyets et al. [125] were one of the first ones to implement this technique by replacing temporal expression words with potential synonyms from WordNet in order to generate additional training examples. Later, a number of authors [126–128] experimented

Augmentation Technique	Category	Sub Category	Generated Sentence
Original			I went to the market and bought some flowers. I had no bags.
EDA	Contextual	Thesaurus	I went to the **store** and bought some **tulips**. I had no bags.
		Embedding	I went to the **bazaar** and bought **a few** flowers. I had no bags.
		Language Model	I went to the **store** and **purchased** some flowers. I had no bags.
	Random	Character-level	I went to the **mrkt** and bought **smoe** flowers. I had no bags
		Word-level	**I to** the market and bought some flowers. I had no bags.
		Sentence-level	**I had no bags.** I went to the market and bought some flowers.
Paraphrase	Machine Translation	Round-Trip	I went to the market and **bought flowers**. I had no bags.
		One-Way	**Je suis allé au marché et j'ai acheté des fleurs. Je n'avais pas de sacs.**
	Controlled Generation	Language Model	I went to the **supermarket** and **purchased** some flowers. I had no bags.
		Generative Model	**I went the market and have bought flowers.**
	Rule-based		I went to the market and **some flowers were bought**. **I'd** no bags.
Compositional Generation	Language Model	Masking	I went to the **mall** and bought some **clothes**. I had no **money**.
		Prompting	I went to the market **to buy a T-shirt from the store.**
	Generative Model		**So I went to go to work carefully**
	Interpolation		He had no bags market and bought some garden.
	Structural		I went to the market and bought some **bags**. I had no **flowers**.

Table 5.1: Examples of generated sentences upon using different augmentation techniques

with word replacement for sentiment analysis and toxic comment classification and reported improvement in model performance. Apart from synonyms, some authors [118, 129] have also used hyonyms and hypernyms as word replacements. Xiang et al. [130] looked into replacement using words having the same POS tag. Their experiment on 8 classification datasets showed improved accuracy in deep learning models.

2. Semantic Embedding-based: Semantic embedding represents a word in a dense vector by making sure that similar words are close to each other in the embedding space. As a result, a number of authors [131, 132] have used pretrained neural word embeddings for word substitution instead of an external thesaurus [133]. Wang and Yang [134] and Li et al. [135] performed word replacement with one of the k-nearest-neighbor words using cosine similarity. The method helped improve model performance in classifying the sentiment of tweets and product reviews. Madukwe et al. [136] used counter-fitted word embedding [137] and skip-gram model [138] for improving hate speech detection. Their

proposed method outperformed the baseline methods on two datasets. Word embedding replacement, however, suffers from lack of context and struggles to fetch synonyms for words with multiple meanings and few synonyms.

3. Language Model (LM) based: Large pretrained LMs can be used to predict synonyms that are not only similar in meaning but also fit the context in principle. Alzantot et al. [139] utilized the Google 1 billion words LM [140] to choose synonyms that have a high probability of fit. Gao et al. [141] computed a weighted average of the embeddings of all possible synonyms predicted by LMs as a replacement. Instead of only relying on synonyms to generate new data, Sosuke Kobayashi [142] made use of context and replaced words in sentences with other words having paradigmatic relations. This was done by modifying a bi-directional LM and making it label-conditional. For example: 'the actors are amazing' gets augmented into 'the performances are fantastic', 'the films are fantastic' and so on.

5.2.2 Random

This category involves simple and easy transformations that are usually context independent. Such techniques are often used to add data perturbations without changing the original label. Models trained on this augmented data are often more robust.

1. Character-level: To make NMT models less susceptible to adversarial examples, Belinkov and Bisk [143] added artificial and natural noise to the training data on a character level. This includes random switching of single letters (cheese → cehese), randomization of the mid part of a word (cheese → ceehse), the complete randomization of a word (cheese → eseehc) and the replacement of one letter with a neighboring letter on the keyboard (cheese → cheeae). Following similar techniques, Feng et al. [144] reported outperforming their baseline model in terms of diversity, fluency, semantic context preservation, and sentiment con-

sistency. Karimi et al. [145] proposed AEDA which includes random insertion of punctuation marks and blanks into the original text. On 5 text classification tasks, model trained on AEDA augmented data outperformed those trained on EDA [36].

2. Word-level: Wei et al. [36] used a combination of random word-level augmentation techniques like random deletion/insertion of a word or swapping two random words in a sentence. Miao et al. [146] and Rastogi et al. [147] implemented similar techniques and achieved performance boost in models for opinion mining and toxic comments classification. Others have looked into replacing non-important words with random words [148] and randomly swapping any two words in a sentence [149, 150] for text classification, and sequence labeling. To make models more robust to common spelling mistakes, a few authors [118, 151] have also introduced a list of the most common English misspellings. For example, replacing "across" as "accross" to generate an augmented text containing a misspelling.

3. Sentence-level: Yan et al. [152] performed random deletion, insertion, and shuffling of sentences in legal documents to increase the training dataset. Similarly, Yu et al. [153] employed an attention mechanism for both word-level and sentence-level random deletion in their proposed hierarchical data augmentation technique for text classification.

5.3 Paraphrase

This technique generates new paraphrases by rewording the original sentence while preserving its meaning.

5.3.1 Machine Translation (MT)

This is a popular and easy to use method to generate paraphrases by making use of machine translation.

1. Round-Trip Translation: Here, the original sentence is translated into some other language using a translation model and re-translated back to the original language([34, 154]). If the new sentence is different from the original sentence, it is added to the training dataset. A number of authors [155, 156] have used NMT-based models to generate paraphrases by translating from English to German/French and back to English and reported improvement in model performance for intent detection and other text classification tasks. To generate more diverse paraphrases, some [157–159] have proposed techniques to control the generated paraphrases using syntactic information and latent variables.

2. One-Trip Translation: For multilingual datasets, a unidirectional approach is taken to generate a paraphrase in a different language. This is especially useful for low resource languages which can be generated by translating rich languages like English or French. Bornea et al. [160] used this technique to augment the original QA English training data with MT-generated data and created a corpus of multilingual QA pairs that was 14 times larger than the original dataset. Once trained on the new corpus, their model outperformed the baselines on multilingual QA datasets. Others [161, 162] have used similar techniques for sentiment analysis and text classification of non-English tweets and answers.

5.3.2 Controlled Generation

Deep learning models are often used to generate text through token prediction techniques. However, during paraphrase generation, the generated text must be controlled in some ways to make sure that the new sentence still preserves the same meaning and semantics.

1. Language Model (LM) based: Paraphrase generation through this technique involves masking tokens from the input and tasking the model with recovering and outputting them in a sequence. Regina et al. [151] randomly masked tokens of the input sentence and used a pretrained BERT to output a probability distribution over the vocabulary for each masked word. A replacement was made if it had the same POS tag as the original word and if the cosine similarity between their embeddings was above a given threshold. The technique significantly improved generalization of machine learning models in low-data regimes. Others [155, 163–165] have used transformer-based models to generate paraphrases for boosting the task of intent classification, question-answering etc.

2. Generative Model based: To produce more diverse paraphrases, a number of authors have taken the advantage of generative models like VAE (Variational Auto Encoder) and GAN (Generative Adversarial Networks). Malandrakis et al. [166] proposed a conditional VAE DA technique which improves model performance in low-data regime for intent classification. Similarly, Cao et al. [167] proposed a conditional GAN based model which uses a diversity loss term to encourage the generator to produce more diverse paraphrases. Likewise, the Diversity-Promoting GAN proposed by Xu et al. [168] assigns low reward for repeated text and high reward for novel text to prompt diverse outputs.

5.3.3 Rule-based techniques

These are easy to implement techniques that usually follow simple if-else rules to construct paraphrases. For example, Coulombe [118] proposed transforming verbal forms from contraction to expansion and vice versa (I've → I have). They also generated paraphrases by changing the active voice of a sentence into passive voice (and vice versa) using a set of transformation rules. Their method achieved very good results in a number of text classification tasks. Similarly, Regina et al. [151] generated paraphrases by abbreviating a group of words or expanding an abbreviation using word-pair dictionaries. Ribeiro et al. [169] also proposed a set of rules to generate paraphrases that were used to perform adversarial attacks on models in order to improve their robustness on a series of tasks (Machine Comprehension, Sentiment Analysis, Visual QA). Rules included transformations like: What VBZ → What's, What NOUN → Which NOUN.

5.4 Compositional Generation

This includes DA techniques that compose new sentences using deep learning models or manipulating sentence structure/feature space. Unlike paraphrases, sentences generated here are label preserving but might not be grammatically correct or carry the same meaning/semantics of the original text.

1. Language Model (LM) based: Techniques here usually follow 3 steps: prepend the class label to each text in the training data, fine-tune a large pre-trained LM on this modified training data (for GPT-2, the fine-tuning task is generation and for BERT, it is masked token prediction) and finally use the fine-tuned LM to generate new samples by providing only the class label (for BERT) or the class label and a few initial words as the prompt for the model (for GPT-2).

    (a) Masking: A certain percentage of tokens are masked and the model is trained to predict the masked tokens by gathering the context from the surrounding tokens. A number of authors have proposed a variation of BERT [111] (C-BERT [170], Aug-BERT [171], BAE[172]) to augment the training data and reported improved model performance on different text classification tasks. Pantelidou et al. [173] further showed that masking selective words (sentiment words) instead of random words before feeding it into BERT improves model performance on movie review datasets. Alongside BERT, Yu et al. [174] also used distil-roBERTa [175, 176] for text generation and reported performance boost in financial sentiment analysis task.

(b) Prompting: Given the first few initial words of the input sentence as prompt and the label of the original sentence, a model is tasked with completing the rest of the sentence by predicting the subsequent tokens. A number of authors [35, 177–179] have reported improvement in model performance on a wide range of text classification tasks upon training the classifier on datasets augmented using guided sentences generated by GPT-2 [102]. Authors like Yoo et al. [180] and Azam et al. [181] have also successfully utilized GPT-3[182] and MT5[181] model for text generation using prompting.

2. Generative Model based: A number of authors [183–186] have used GAN and VAE to generate new sentences that are coherent and have higher quality. Frédéric et al. [187] received a performance boost in binary classification tasks upon using a separate VAE per class to generated new data via random sampling of the latent space. Others have proposed variations like RELGAN [188], GAN+ [189], VGAN [190] etc. for natural sounding text generation. Shehnepoor et. al. [191] used ScoreGAN to generate reviews with specific semantics by incorporating review text and review ratings into the loss function and reported major improvement in model accuracy in fraud review detection task.

3. Structural: Such transformation composes new sentences by utilizing certain features of an existing sentence structure like dependency tree, POS tag, grammar etc. Motivated by image cropping and rotation, Sahin and Steedman [192] proposed swapping (rotation) or deleting (crop) the children of the same parent. Similarly, Louvan et al. [193] generated smaller sentences by cropping fragments of the dependency tree. They also rotated target fragment around root of the dependency parse tree and produced new sentences. Such techniques

improved model performance in low resource slot filling and intent classification. Subject/object inversion by Min et al. [194] also yielded higher generalization capability of model in NLI.

4. Interpolation: In numerical analysis, interpolation is the process of constructing new data points from existing points [195]. For text DA, this can be interpreted in the feature space where given two data-label pairs, virtual data-label pairs are created through linear interpolations of the pair of data points.

   (a) MixUp: Mixup trains a neural network on convex combinations of pairs of examples and their labels. Inspired by Zhang et al. [196] who combined two random images in a mini-batch in some proportion to generate synthetic examples, Guo et al. [197] proposed wordMixup and sentMixup for text data. First, they zero-pad two random sentences to the same length and either combine their word embeddings in some proportion directly (wordMixup) or pass the word embeddings through an encoder and then combine their last hidden state sentence embeddings in a certain proportion (sentMixup). On 5 text classification tasks, this technique improved the accuracy rate of CNN and LSTM models. Instead of hidden vectors, Yoon et al. [198] applied Mixup on input text. Their method outperformed previous hidden-level Mixup methods on multiple NLP tasks.

   (b) SMOTE: Synthetic Minority Oversampling Technique [199] is used to fix class imbalance by generating minority class examples using interpolation [200]. Unlike Mixup, only instances of the same class get interpolated here. Curukoglu et al. [201] proposed SMOTE-text for TF-IDF vectorization by integrating Turkish dictionary for oversampling during text processing and classification. Wang et al. [202] improved the performance of SVM classifier on the imbalanced patent document dataset using P-SMOTE which focuses on the blank spaces along positive borderline of SVM and generates pseudo positive examples. Others have achieved good results by using a variation of SMOTE in classification tasks like detection of toxic comments [203], emotions [204] and sentiment of scientific citations [205].

5.5 DA methods for NLP Tasks

In this section, we give a brief overview of the DA methods that have been applied on a wide-range of NLP tasks.

1. Text Classification: It is one of the most popular NLP tasks which involves assigning a label or class to a given text. Tasks like sentiment analysis, fraud review detection, and news categorization fall under this category. Most of the DA techniques discussed in the previous section has been or can be extensively used for text classification. More authors like Ren et al. [206], Wei et al. [207], and Liu et al. [208] have proposed different DA techniques and reported boost in classification accuracy on tasks like irony recognition, offence detection, and question type classification.

2. Inference: It is the task of determining whether a 'hypothesis' is true (entailment), false (contradiction), or undetermined (neutral) given a 'premise'. Min et al. [209] applied a number of techniques like Inversion, Passivization, and Random shuffling to generate augmented data which improved BERT model performance on the MNLI corpus. Singh et al. [210] proposed XLDA, a cross-lingual DA method that replaces a segment of the input text with its translation in another language. Training with XLDA achieved state-of-the-art performance for Greek, Turkish, and Urdu language on the XNLI dataset.

3. Paraphrase Detection: Given two sentences, the task involves determining whether they have the same meaning. This is helpful for plagiarism detection and duplicate question identification. Shakeel et al. [211] generated additional paraphrase (using reflexivity, symmetry, transitive extension) and non-paraphrase

(using symmetry, non-paraphrase extension) pairs and improved their model performance for paraphrase detection. Likewise, Anchiêta et al. [212] used a back-translation strategy to balance the training dataset for paraphrase detection in Portugese. Some authors [213, 214] have also applied DA techniques like rearrangement, back-translation and segment reordering to improve model performance on Semantic Textual Similarity task.

4. Grammatical Error Correction (GEC): It is the task of correcting different errors in text like spelling, punctuation, grammatical and word choice errors. Given a potentially erroneous sentence as input, a GEC system is expected to transform it to its correct version. Xu et al. [215] used a combination of concatenation, misspelling, substitution, deletion, and transportation to generate erroneous data to train their transformer based GEC model and made it more robust. Others have looked into editing latent representations of original sentences [216] as well as using error patterns and POS tags [217] to generate synthetic data to improve performance of GEC models.

5. Neural Machine Translation (NMT): It is the task of correctly translating sentences from one language to another. Nguyen et al. [218] proposed training multiple models on both backward and forward translation tasks and then using them to generate data from both lingual sides. The technique achieved a boost in BLEU score on WMT'14 English-German and English-French translation tasks. For cross-domain NMT, Peng et al. [219] proposed generating a large-scale pseudo in-domain (IND) parallel corpora using IND dictionaries and Out of Domain bi-text. Li et al. [220] showed that exposing model to bad segmentation during training can improve robustness on the IWSLT English to German dataset.

6. Summarization: It is the task of producing the summary of one or many documents. Parida et al. [221] proposed a DA technique where given a summary, the

model generates the text. The synthetic text-summary pair is then merged with the original data to train the model on text summarization for German language and it improved on low resource setting. Zhu et al. [37] proposed DA for query-focused summarization. They used body text of citation as document, article title and section titles to form query, and statement as the summary. Fabbri et al. [222] fine-tuned pretrained models using pseudo-summaries produced from Wikipedia data containing characteristics of target dataset. This helped models achieve SOTA zero-shot abstractive summarization performance.

7. Question Answering (QA): It is the task of retrieving the answer to a question from a given text. Asai et al. [38] generated synthetic data by converting a question into an opposite one by replacing words with their antonyms or adding/removing negation words. The corresponding answer was obtained the same way. The method improved SOTA model performance on three QA datasets. Yang et al. [223] generated positive and negative data by using passage retrieval and later finetuned BERT models for open domain English and Chinese QA datasets. Riabi [224] used DA for cross-lingual QA models by translating the SQUAD monolingual corpus. Their method achieved SOTA results on four multilingual datasets.

8. Sequence Tagging: POS tagging is a popular sequence labeling task which marks up a word in a text corresponding to a particular POS. To improve POS tagging of ancient Chinese texts, Shen et al. [225] used SikuRoberta [226] to generate synthetic text by randomly masking verbs and entities in the training sentence and then used a tagger model to label the generated data. Vania et al. [227] also applied DA techniques to improve parsers by generating synthetic sentences via dependency tree morphing [192] and nonce sentence generation [228]. Another popular sequence labelling task is Named Entity Recognition (NER) which involves detecting the entities in the text. Chen et al. [229] proposed an

instance-level and feature-level DA technique that improved the performance of Glove and BERT based models for NER in low-resource setting.

9. Task-Oriented Dialogue Systems: These include conversational agents that help users accomplish a task by identifying the domain, determining the intent and filling the slots from the conversation. Gao et al. [230] used a paraphrase generation model to generate additional user utterances which improved the task completion rate of a dialogue system especially in low resource setting. Others have used DA techniques to improve Natural Language Generation (NLG) which converts structured meaning representation (MR) to NL. Example: Xu et al. [231] generated synthetic MR annotations consisting of an intent and slot value pairs from open-domain texts. They combined a self-trained neural retrieval model with a few-shot learned NLU model for this. Some have also observed improved model performance using DA methods for dialogue state tracking [232, 233].

5.6 Experiments and Discussion

In this section, we apply some of the popular text augmentation techniques on a number of well-known NLP tasks and analyse their significance. The DA techniques include Random Deletion, Random Insertion, Random Swap, Synonym Replacement, EDA, Back Translation, Masking with BERT, and Prompting with GPT-2. We use Kumar et al.'s [35] implementation of these techniques with slight modifications based on the task at hand. We test them on a variety of benchmark datasets for a wide range of NLP tasks like (i) Text Categorization using BBC News [234], (ii) Sentiment Analysis using IMDB Movie Reviews [235], (iii) Emotion Detection using CARER [236], (iv) Dialogue Slot Filling and Intent Detection using SNIPS [72].

Like Kumar et al., to investigate how effective DA techniques are in the face of data scarcity, we artificially simulate a low data regime. For each dataset, we experimented once with 10 labeled data points per class and second time with 100 labeled data points per class. We generate one augmented text for each original text, doubling the size of the initial data points (with the exception of EDA which triples it). We also experimented with the whole dataset without augmentation for better comparison. We use the pretrained BERT-base [112] model as our base classifier and use the same hyper-parameters across all datasets and techniques. The BERT-base model has 100M parameters. Like Kumar et al., we also use a learning rate of 2e-5, and dropout ratio of 0.1 for different augmentation methods. We use accuracy as the evaluation metric for all the NLP tasks except for Slot Filling (which uses F1-score). Because the performance can be heavily dependent on the specific data points chosen [237], for each dataset, we sample labeled data from the original dataset 15 times to form 15 different training sets, and report the average result. Table 5.2 shows the accuracy and F1 score obtained by the Bert-base classifier upon applying different augmentation techniques on low resource data setting.

The BBC News dataset has 5 news categories (labels) namely: business, entertainment, politics, sports, technology with approx. 1000 train and 224 test examples. Our Bert-base classifier achieves an accuracy rate of 95.98% on the whole dataset without any augmentation. However, in our artificially simulated low data regime, upon training the classifier on only 50 examples (10 per label), the accuracy drops to 78.92%. Upon applying DA, it is seen that almost all of the techniques achieve an accuracy of approx. 90% (exception: Prompting using GPT-2). Among the implemented techniques, Random Swap seems to achieve the highest accuracy (91.96 %). What is surprising however is the fact that on the second setting with 100 examples per label (500 train examples total), the classifier achieves an accuracy of 94.315% without any data augmentation. This performance is very close to when the classifier is trained on the entire dataset. For news categorization, it seems as though text data augmentation techniques are redundant unless dealing with an extreme data scarcity. This finding aligns with what was observed by the previous authors [124, 238]. Similar case is observed with the IMDB Movie Review dataset which has two labels, positive and negative, for approximately 36000 training examples and 7500 test examples. When trained on the entire training data, the classifier achieves an accuracy of 85.71%. When trained using only 20 examples (10 per label), an accuracy of 52.88% is obtained without any augmentation. With augmentation, accuracy of around 53% to 54% is achieved, the highest being 54.78% with EDA. This increases to 70.56% when EDA is applied to augment a training dataset of only 200 examples (20 per label). Without augmentation, it is around 67.19%. Although improvement is noticed upon applying DA, it cannot be deemed as too significant. Next is the CARER dataset which is an emotion dataset with 6 different emotions (anger, fear, joy, love, sadness, surprise) collected from Twitter using hashtags. It has approx 16,000 train and 2000 test examples. Our BERT-base model achieves an accuracy of 92.85% when trained on the whole dataset. However, when only 60 examples (10 per label) are used, the accuracy drops to approx. 16.83%. Upon using augmentation techniques, it is improved to around 20% to 25%, the highest being 26.396% using Random Swap. A significant improvement is observed on the second setting (100 per label) upon using EDA, which improves the accuracy to 64.15% whereas without any augmentation, it was just 32.33%.

Lastly, we use the SNIPS dataset for intent detection and slot filling tasks. It has several crowd-sourced queries of various complexity distributed among 7 user intents (SearchCreativeWork, GetWeather, BookRestaurant, PlayMusic, AddToPlaylist, RateBook , SearchScreeningEvent) and 74 slots. The complete dataset has approx. 13,000 train and 700 test examples. We perform intent detection and slot filling simultaneously by following the framework proposed by Chen et al.[239] using our Bert-base classifier. When trained on the whole dataset, our classifier achieves an accuracy of 97.7% for intent detection and an F1-score of 94.7% for slot filling. For slot filling task, a lot of the data augmentation techniques were not applicable given the BIO tags associated with each word in the sentence. This is why we only used Random

Deletion (deleting a word and its corresponding BIO tag), Random Swap (swapping words and their corresponding BIO tags), Synonym Replacement (ensuring a word is replaced by a one-word synonym to preserve the BIO tags) and EDA (a combination of the previous three methods) here. When only 70 train examples (10 examples per intent) were used, the accuracy and F1-score dropped to 75.4% and 0% respectively. This highlights the fact that a lot more training data is required to train a classifier when it comes to slot filling. Upon applying data augmentation techniques, accuracy and F1-score were improved to 90.7% and 38.3% using EDA which is much better especially for slot filling. On the second setting, without any augmentation, the classifier already achieves an accuracy and F1-score of 96.5% and 81.7%. Upon applying augmentation techniques, not much improvement is noticed for accuracy (96.7% using random swap). F1 score for slot filling, however, increases to 87.3% using random deletion.

From these experiments, we can conclude that, DA techniques are definitely useful in improving classifier performance especially in low data regimes. Moreover, how much improvement will be observed, depends a lot on the type of NLP task at hand, the quality of the dataset, and the number of labels. Furthermore, simple EDA technique seems to outperform the rest in most cases. One of the biggest reasons for this, however, might be because of the fact that the dataset created using EDA is 1.5 times larger than the size of the dataset generated using other augmentation techniques. Given the promising results of DA, we decided to apply them to improve the performance of our dialog act classifiers.

Task	Dataset	Data Info	No Aug	Random Delete	Random Insert	Random Swap	Synonym Replacement	EDA	Back Translation	BERT	GPT2
Text Categorization	BBC News	10	78.92 ± 5.85	90.35 ± 2.72	91.07 ± 1.95	**91.96 ± 1.92**	90.77 ± 2.89	90.71 ± 2.18	90.92 ± 2.63	91.54 ± 2.06	88.92 ± 2.68
		100	94.31 ± 1.39	94.49 ± 0.97	94.31 ± 1.30	94.97 ± 1.01	94.94 ± 0.841	**95.26 ± 1.50**	94.96 ± 1.03	94.64 ± 1.08	93.48 ± 1.33
Sentiment Analysis	IMDB Movie Review	10	52.88 ± 3.20	53.84 ± 3.76	54.04 ± 3.93	54.52 ± 3.51	54.43 ± 3.51	**54.78 ± 3.48**	53.56 ± 2.92	53.34 ± 4.03	53.19 ± 3.28
		100	67.19 ± 3.87	67.64 ± 4.34	68.83 ± 4.66	67.66 ± 5.79	66.49 ± 5.68	**70.56 ± 3.78**	68.32 ± 4.37	67.87 ± 7.07	60.38 ± 4.57
Emotion Detection	CARER	10	16.83 ± 5.23	25.03 ± 6.74	23.03 ± 7.79	**26.39 ± 7.43**	23.05 ± 4.82	24.47 ± 7.63	24.49 ± 6.19	22.77 ± 8.16	20.34 ± 6.09
		100	32.32 ± 9.42	44.58 ± 10.47	44.43 ± 7.48	44.27 ± 10.13	44.97 ± 9.07	**64.14 ± 8.07**	44.58 ± 10.47	44.71 ± 9.26	28.7 ± 9.79
Dialogue Slot & Intent	Snips (Intent Detection)	10	75.41 ± 2.50	81.68 ± 2.32	-	81.39 ± 3.17	80.25 ± 2.58	**90.71 ± 1.85**	-	-	-
		100	96.50 ± 0.30	**96.73 ±0.42**	-	96.65 ± 0.38	96.51 ±0.37	96.63±0.40	-	-	-
	Snips (Slot Filling)	10	0.0	3.73 ± 3.36	-	2.67± 3.02	2.71 ± 2.89	**38.30 ± 3.09**	-	-	-
		100	81.72±0.91	**87.33 ± 0.94**	-	83.08 ± 0.71	86.53 ±0.75	87 ± 0.61	-	-	-

Table 5.2: Average Accuracy (for the rest) and F1 score (for slot filling) with Standard Deviation of BERT-base classifier after implementing a wide range of data augmentation techniques for different NLP tasks

Chapter 6

Text Data Augmentation in Dialogue Act Detection

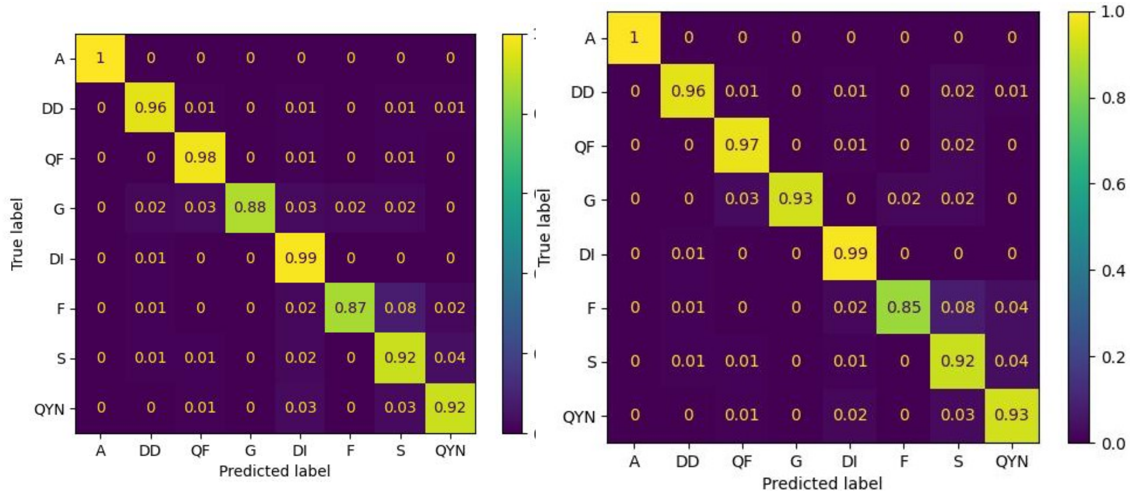
One of the main foci of our work is looking into the applicability of data augmentation techniques for dialog act detection. Although our DA dataset has adequate examples to achieve SOTA results, this is not a common phenomenon. In fact, when working with a low resource language or in a new domain, it is often difficult, expensive and time-consuming to curate enough examples. To tackle this, we investigate the success of augmentation techniques in improving the performance of DA classifiers in two phases. First, we experiment with boosting the accuracy of the baseline SVM model by augmenting the original training dataset. Second, we perform similar experiments for our proposed DA classifier but in a simulated low data regime. This section discusses some of these experimentation and subsequent results in details.

6.1 Experimentation with Baseline

From the experimental results shared in Chapter 4, we can see that the SVM baseline had low accuracy rates for two of the minority classes, Greeting and Feedback, as well as two of the majority classes, Statement and Yes/No Question. From our earlier experimental results from Chapter 5, we decided to use EDA and Back-translation techniques to create new data from the existing train examples for four of these classes and see if the baseline accuracy rate for those classes improve with the addition of newly

Data Augmentation Method	Greeting	Feedback	Statement	Yes/No Question
None	239	329	3250	3384
EDA	421	713	6460	6859
Back-translation	403	647	5968	6301

Table 6.1: Number of train examples of four classes before and after applying data augmentation techniques



(a) Without Augmentation

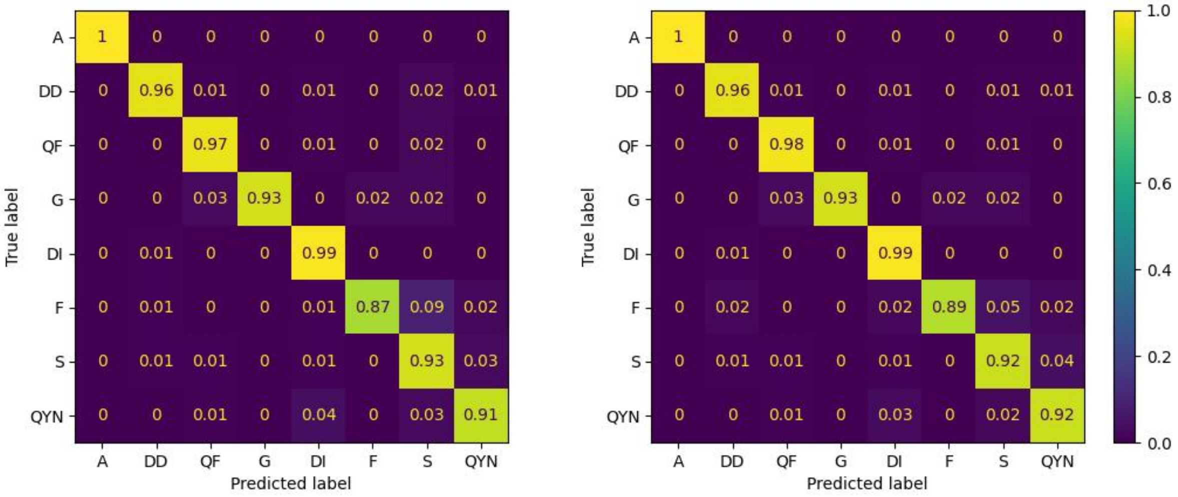
(b) With EDA on four classes (G, F, S, QYN)

Figure 6.1: Confusion matrices of the baseline SVM trained on a dataset without augmentation and with augmentation (EDA)

generated data. For both EDA and Back-translation, we use more or less the same experimental setup mentioned in Chapter 2 to generate new examples. Since EDA uses a culmination of 4 techniques (Random Insertion, Random Deletion, Synonym Replacement, and Random Swap), it creates a dataset larger than Back-translation. There were also a number of newly generated examples that were exactly the same as some of the original ones. We made sure not to include those. We trained out SVM classifier on the newly generated dataset and ran it on our original test dataset. In other words, our test dataset remained the same in all the experiments. Table 6.1 provides more information on the number of examples per class upon augmentation.

At first we experiment with EDA and discuss its results and findings. Figure 6.1

shows two confusion matrices: the first one is generated after training the baseline on the original training dataset (without data augmentation) and the second one is generated after training it on the dataset augmented using EDA on the previously mentioned four classes . It is observed that, when the number of training examples for the minority class Greeting is doubled (increased to 421 from 239), the accuracy rate jumps from 88% to 93%. Slight improvement (a jump from 92% to 93%) is also observed for the majority class Yes/No Question when the number of training examples is made double to 6859 from 3384. However, for the remaining two classes, Feedback and Statement, no improvement is observed despite doubling the size of their training examples. In fact, some of the new data generated using EDA caused the accuracy rate for Feedback class to drop from 87% to 85%. One of the main reasons for this might be attributed to the poor quality of the generated data for some of these classes which might not have been label preserving. For example: new training sentences like ‘assistant name you do have brothers or sisters’ and ‘confirmed supreme court justices have to be do’ that were generated for the class Yes/No Question using EDA are wrong and might have contributed to the decline in performance. As for the Statement class, we assume that, when more augmented data is included, it gets larger and completely over powers the Feedback class. As a result, a number of feedbacks are then misclassified as statements. This assumption is further proven by the confusion matrix in Figure 6.2 where we applied EDA on three classes: Greetings, Feedback, and Statement. We can see that the number of Feedback examples being misclassified as Statement increases to 9% from 8%. Moreover, the overall accuracy of the SVM classifier drops to 95%. Based on these observations, we decided to only apply data augmentation techniques on the two minority classes of our dataset: Greeting and Feedback with the hopes of improving the overall performance of the baseline SVM. From the confusion matrix in Figure 6.2, we can observe that the accuracy rate jumps from 87% to 89% for Feedback class and from 88% to 93% for Greeting class. Despite this, the overall accuracy remains the same at 96% and so does the macro average of



	A	DD	QF	G	DI	F	S	QYN
A	1	0	0	0	0	0	0	0
DD	0	0.96	0.01	0	0.01	0	0.02	0.01
QF	0	0	0.97	0	0.01	0	0.02	0
G	0	0	0.03	0.93	0	0.02	0.02	0
DI	0	0.01	0	0	0.99	0	0	0
F	0	0.01	0	0	0.01	0.87	0.09	0.02
S	0	0.01	0.01	0	0.01	0	0.93	0.03
QYN	0	0	0.01	0	0.04	0	0.03	0.91

(a) With EDA on three classes (G, F, S)

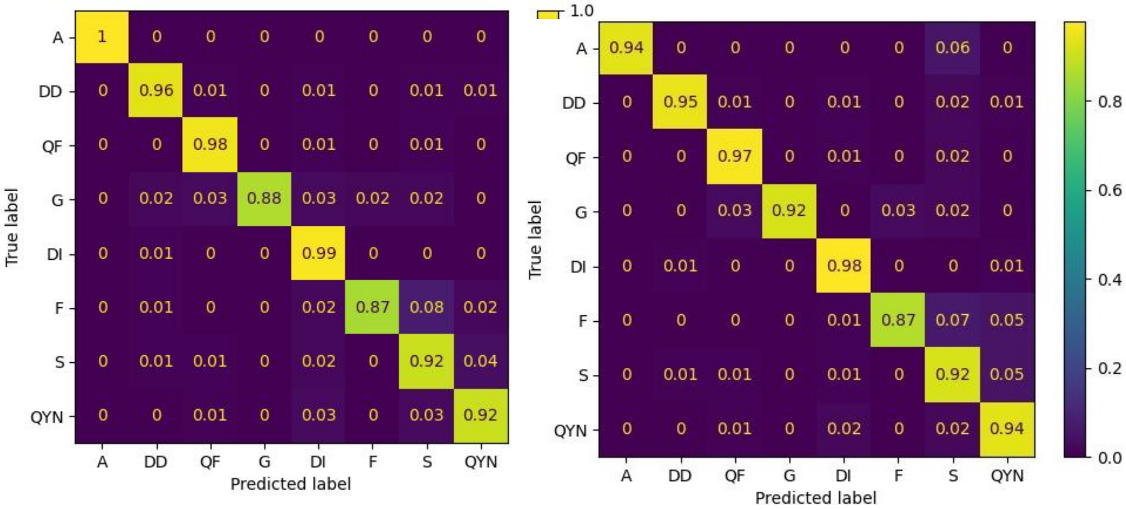
	A	DD	QF	G	DI	F	S	QYN
A	1	0	0	0	0	0	0	0
DD	0	0.96	0.01	0	0.01	0	0.01	0.01
QF	0	0	0.98	0	0.01	0	0.01	0
G	0	0	0.03	0.93	0	0.02	0.02	0
DI	0	0.01	0	0	0.99	0	0	0
F	0	0.02	0	0	0.02	0.89	0.05	0.02
S	0	0.01	0.01	0	0.01	0	0.92	0.04
QYN	0	0	0.01	0	0.03	0	0.02	0.92

(b) With EDA on two classes (G, F)

Figure 6.2: Confusion matrices of the baseline SVM trained on a dataset augmented using different classes (EDA)

precision (96%) and F1-score (95%). However, the macro average of recall jumps from 94% to 95%. This means that the baseline classifier can now predict more examples of each class correctly. For example: sentences like ‘I want to see them beg for me to stop hurting’, ‘give hindu temples what happened to them a 5 out of 6 stars’, and ‘see ya’ that were formerly misclassified as Indirect Order, Factual Question, and Direct Order have now been correctly classified as Statement, Direct Order, and Greeting. Moreover, the misclassification of feedbacks as statements reduces to only 5% from 8%.

Similar results are observed when we use Back-Translation instead of EDA for augmenting the dataset. For example, in Figure 6.3 we can see that doubling the number of training examples for the class Greeting, Feedback, Statement, and Yes/No Question using BT only improves the accuracy rate for Greeting (from 88% to 92%) and Yes/No Question (from 92% to 94%). Like EDA, the accuracy rate for Feedback and Statement remains more or less the same even after augmentation. It is to be noted that, in case of Back-translation, especially when done in between two high

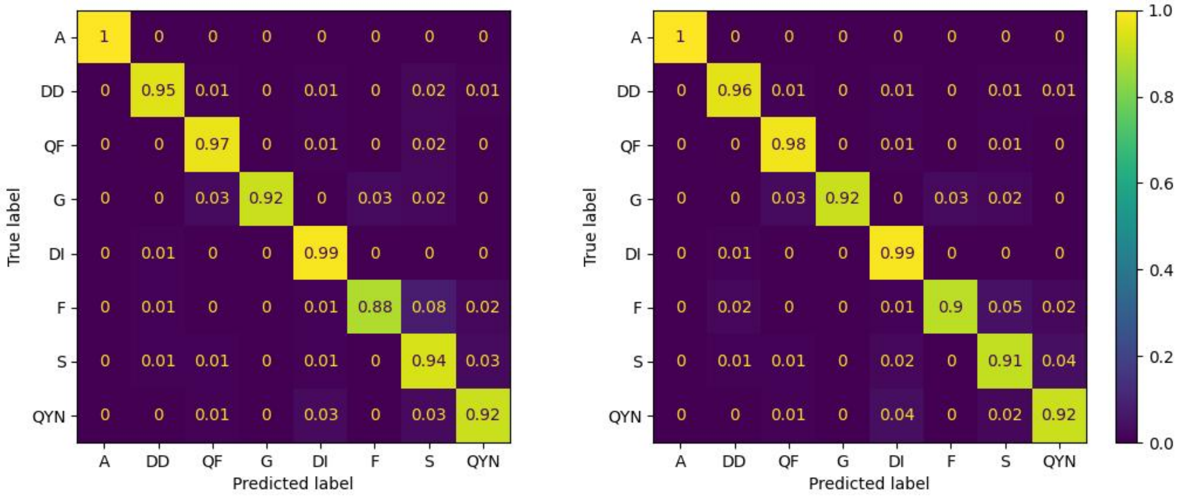


(a) Without Augmentation

(b) With BT on four classes (G, F, S, QYN)

Figure 6.3: Confusion matrices of the baseline SVM trained on a dataset without augmentation and with augmentation (BT)

resource languages (English to German and back to English in this case) opens up the possibility of getting more accurate translation. But the downfall to this is that some of the re-translated texts are the same as the original sentence and so they could not be included in the new training set. In Figure 6.2, when we apply BT on three classes: Greeting (G), Feedback (F), and Statement (S), we do observe slight improvement in accuracy rate for Feedback (from 87% to 88%) and for Statement (from 92% to 94%). However, the percentage of feedback examples being misclassified as statement remains the same (as high as 8%). When BT is applied only on the two minority classes, Greeting and Feedback, the accuracy of Feedback class jumps to 90% which is incredible. Moreover, the number of feedback examples being misclassified as statement reduces to only 5%. However, the inclusive of these augmented data causes a slight drop in accuracy rate in the Statement class (91%). Despite all this, unlike EDA, the overall accuracy and macro average of precision, recall and f1-score of the baseline when trained on data augmented using BT remains the same at 96%, 96%, 95% and 95% respectively.



(a) With BT on three classes (G, F, S)

(b) With BT on two classes (G, F)

Figure 6.4: Confusion matrices of the baseline SVM trained on a dataset augmented using different classes (BT)

From these observations, we can draw the conclusion that data augmentation techniques work the best for minority classes (i.e., classes with a very low number of examples). By increasing the train dataset with newly generated examples, we can improve the accuracy rate of dialog act classifiers especially for minority classes. However, a lot of the improvement depends on the quality of the generated data. This is because, given a task, not all data augmentation techniques will generate examples that are of high quality (i.e., label preserving). Thus, when implementing such techniques, the data quality and the percentage of label preserved should be taken into consideration.

6.2 Low Resource Setting

To harp on the argument that data augmentation techniques work the best in the face of data scarcity (i.e., when the number of examples per label is very small) we carry out another batch of experiments but this time for our proposed DA classifier. Here, we simulate a low resource data setting by taking only 10 examples for each class. Then we apply different augmentation techniques to generate one (in case of

BT, SR, RS, RI, RD) or more (in case of EDA) new sentence(s) from every original sentence. Like the last time, we make sure to omit newly generated sentences that are exactly the same as the original sentence. Thus, a new training dataset is created by including the newly generated unique data points to the randomly sampled small training dataset with 80 examples (10 per label). We later train our proposed BERT-base model on this augmented training dataset. The test dataset, however, remains the same as always. After training, we want to test whether the performance of our classifier in detecting dialog intents improves in the face of data scarcity. We repeat the experiment 15 times and report the average accuracy with standard deviation just as discussed in Chapter 5. We also repeat the same experiment but with 100 examples per label the second time. These experiments are of utmost importance especially in the real world. Often times it is difficult to curate enough examples for every class. Synthetically generating new data points from the existing examples can help increase the training dataset with ease. But whether it will improve the performance of the classifier in the task of user act detection is what we want to investigate. Table 6.2 provides a brief summary of our experimental results. We first take 10 examples per class and train our BERT-based model. When ran on the original test dataset, it achieves an accuracy of 76.012 ± 7.606 %. However, when the model is trained on the augmented training dataset, the performance can be improved significantly with an accuracy ranging between 79.559 ± 4.002 % to 83.812 ± 6.226 %. Moreover, the model performance becomes very stable (indicated by low standard of deviation) and achieves an average accuracy of 83.641 ± 4.477% when it is trained on the dataset augmented using a technique as simple as Synonym Replacement. On the flip side, when training data augmented using Back Translation is used to train our classifier, unlike the rest of the techniques, a drop in performance is observed (accuracy rate becomes as low as 74.465 ± 8.09 %). The most plausible reason for this might be that the new data generated using this method is of poor quality and might not be label preserving. Upon further analysis, we in fact noticed that bad translation had led to a number of newly generated sentences having a wrong label. For example: Yes/No Questions like 'has tampa ever been hit by a hurricane', 'can an infinite geometric series have a sum' upon undergoing Back Translation generate 'tampa ever has been hit by hurricane' and 'an infinite geometric series can have a sum' which are no longer examples of Yes/No Questions and yet have been tagged as one. This might have led to the ultimate drop in performance. Another thing to note here is that, when the number of examples per label is so small, the fluctuations in accuracy rate is very high from 4 to 8%. As a result, repeating the experiment 15 times gives us a better idea on the overall expected performance of the model by taking the quality of the sampled training examples out of the equation.

Next, we take a look at what happens when we have a dataset with only 100 examples per class. Although small, it definitely is a lot larger than having only 10 examples per class. Without augmentation, our BERT-based model achieves an accuracy of 88.935 ± 5.655 % which is not bad. However, when trained on augmented data, the performance improves significantly and reaches above 95% in almost all the cases (exception: Back Translation) which is very impressive. The highest accuracy is obtained when the model is trained on dataset augmented using Masked Token Prediction via BERT (96.443 ± 0.605%). Another important thing to note here is that, in this setting, the performance boost obtained by our classifier remains more or less consistent (as indicated by the low standard of deviation), which means that unlike the last time, the classifier improvement is not highly dependent on the quality of the randomly sampled training data. In fact, for most of these data augmentation techniques, the newly generated data is able to help our classifier achieve a high accuracy that is consistent throughout the 15 rounds of random sampling which is astounding. However, like last time, Back translation does not help much with the model performance for the same reasons. Thus, from these experiments we can confidently say that for dialogue act classification, almost all the augmentation techniques can successfully boost model performance in low data regime. Out of them, EDA,

Data Info	No Aug	Random Delete	Random Insert	Random Swap	Synonym Replacement	EDA	Back Translation	BERT	GPT2
10	76.01 ± 7.6	81.59 ± 6.35	80.58 ± 5.92	79.55 ± 4	**83.64 ± 4.47**	80.89 ± 5.27	74.46 ± 8.09	82.87 ± 4.17	**83.81 ± 6.22**
100	88.93 ± 5.65	95.22 ± 1.66	96.23 ± 0.6	95.93 ± 0.91	96.18 ± 0.65	96.22 ± 0.49	87.68 ± 10.11	**96.44 ± 0.6**	96.07 ± 1.08

Table 6.2: Average accuracy with standard of deviation of our proposed classifier after implementing a wide range of data augmentation techniques

Masking and Prompting are the top contenders with Back Translation being the one to avoid. When it comes to using data augmentation techniques using large language models, one thing to note here is the risk of introducing linguistic conformity in the training data. So, care must be taken to make sure the augmented data does not unintentionally induce social biases and stereotypes [240].

6.3 Data Augmentation Techniques for DA Classification in French

One of our future goals involve making both of our chatbots multilingual. For this, our DA classifier should be able to identify dialog acts from user utterances in multiple languages, not just English. Because French is the second most popular language among our users (right after English), we decided to use our proposed DA classifier to detect the 8 dialog acts from French utterances. This section discusses in details the process of building a small French dataset, expanding it using augmentation techniques and finally training our DA classifier on it. Although we specifically deal with the French language here, the steps and techniques mentioned can be replicated for detecting dialog acts in any language.

6.3.1 French Dataset Creation

We decided to translate our English dataset into French for the purpose of training and evaluating our DA classifier. Given how large our dataset is, we first looked into translating it automatically using the Google Translate API. However, upon analyzing the translated French sentences, we noticed that most of them were of poor quality. A number of these French sentences were incomplete and did not retain the

Dataset	Total	Apology	Direct Order	Factual Question	Greeting	Indirect Oder	Feedback	Statement	Yes/No Question
Train	234	28	28	27	36	27	31	26	31
Test	150	9	20	20	20	20	20	20	20

Table 6.3: Summary of the sentences translated from English to French for creating the French training and test dataset

meaning of their original English sentences. As a result, some of the newly translated sentences were carrying labels that were no longer true for them. For example: when the English sentence ‘find the cartel vol 2 novel’ with the label Direct Order was automatically translated, we got ‘trouver le cartel vol 2 roman’ in French which no longer conveys the same dialog act. A DA classifier trained and tested on this faulty dataset will surely not give accurate results. To mitigate this issue, we decided to randomly choose a few English sentences per label from our original dataset and then manually translate them into French. For translation, we took the help of two MIRA team members who are fluent in both English and French. For generating the French train dataset, we randomly chose 40 sentences per label for translation. However, our translators at times included two versions of French translations for each English sentence in order to make the French train dataset much larger. Likewise, for generating the test dataset, we randomly chose around 20 examples per label that are completely different from the ones chosen for training. Table 6.3 gives a run down of the distribution of the translated sentences for each dialog act.

6.3.2 Experimentation with BERT and m-BERT

Our proposed DA classifier is based on BERT which is pretrained on monolingual English dataset. Keeping that it mind, the preprocessing of our French dataset was handled a bit differently. For example: we mapped all the French accents to their English counterparts (eg: à → a, â → a, è → e, æ → ae) to make sure the model recognizes all the symbols. Moreover, instead of removing every single punctuation mark, we kept the apostrophes and the hyphens intact so that the meaning of the

Total Number of Epochs	Average Accuracy	
	fine-tuned monolingual BERT	fine-tuned multilingual BERT
3	41	68
6	65	78
10	68	80
20	69	80

Table 6.4: Comparison between the average accuracy obtained by fine-tuning BERT and m-BERT on our French dataset in different epochs

French words do not change. The rest of the experimental setup for our French DA classifier is more or less the same as discussed in Chapter 4. Upon training our classifier on this data for 3 epochs, we obtain an average accuracy of 41%, which is not ideal. Given how small the French training dataset is in comparison to our English dataset, we decided to increase the number of epochs during training to help the neural network learn the structure of the data. As expected, just by doubling the number of epochs from 3 to 6, the accuracy jumps by 24 points and becomes 65%. We experiment further by increasing the number of epochs during training and report the results in Table 6.4). Fine-tuning BERT with a small dataset can often lead to instability in model performance [241]. As a result, we repeated each experiment 5 times and reported an average accuracy of our model to get a better picture. From the Table, it is seen that, when the number of epochs during training is 20, an average accuracy as high as 69% is achieved which is not bad at all since our DA classifier was never pretrained on French corpus.

Given how different French language is from English, we speculate whether using a model pretrained on large French corpus will yield better results. We put our hypothesis to the test by deciding to fine-tune m-BERT or multilingual BERT [112] on this dataset. Like the original English BERT model, m-BERT is a 12 layer transformer but instead of being trained only on monolingual English data with an English-derived vocabulary, it is trained on the Wikipedia pages of 104 languages with a shared word piece vocabulary including French. Using the same experimental set-up and repeating each experiment 5 times, we achieve the results shown in table 6.4. As expected, upon using model pretrained on the French corpus we were able to significantly improve the accuracy rate of our French DA classifier by 10% with the highest accuracy being 80%. This is impressive given how small how fine-tuning dataset is. As future work, it might be interesting to observe how BERT models pretrained on multi-lingual corpus will perform once it's fine-tuned on a multi-lingual DA dataset.

6.3.3 Application of Augmentation Techniques

With the aim of further improving the performance of our French DA classifiers, we decided to implement the augmentation techniques discussed in Chapter 5 to increase the size of our training dataset. However, due to the change in language, some modifications were made to each of the methods. For example, for Random Insertion and Synonym Replacement, we used a French thesaurus instead of an English thesaurus in order to get a list of synonyms of particular French word(s) in a sentence. Similarly, we applied the Back Translation technique by translating French sentences to English and back to French. For methods like Masked Token Prediction using BERT and Prompting using GPT-2, we looked for models that are trained on French corpus for optimal performance and finally decided to use 'bert-base-multilingual-cased' [242] and 'gpt2-wechsel-french' [243] as substitutes. The former is a BERT model pretrained on 104 languages (including French) with the largest Wikipedia using a masked language modeling (MLM) objective, while the latter used the WECHSEL technique (effective initialization of subword embeddings for cross-lingual transfer of monolingual language model) to transfer the English GPT-2 model to four languages (French, German, Chinese, and Swahili). Given that our dataset is already very small, we did not have to simulate a low resource setting like last time through random sampling. Instead, we used the entire French training dataset and augmented

		No Aug	Random Delete	Random Insert	Random Swap	Synonym Replacement	EDA	Back Translation	BERT	GPT2
No of Examples		234	351	457	461	451	430	398	460	462
Model	BERT	68	68	**72**	70	71	67	71	68	**72**
Model	m-BERT	80	81	**84**	82	81	81	82	81	81

Table 6.5: Average accuracy of fine-tuned BERT and m-BERT on our French dataset after implementing a wide range of data augmentation techniques

Technique	Example
No Augmentation	le cortex préfrontal fait-il partie du lobe frontal
Random Delete	le cortex préfrontal du lobe frontal
Random Insert	monsieur le cortex préfrontal fait-il partie du lobe frontal
Random Swap	le cortex préfrontal fait-il lobe du partie frontal
Synonym Replacement	le cortex préfrontal fait-il contribution du lobe frontal
EDA	le aboyer cortex préfrontal fait-il partie du lobe frontal
Back Translation	est la partie du cortex préfrontal du lobe frontal
BERT	le cortex prefrontal fait pour partie du lobe frontal
GPT-2	le cortex prefrontal et le lobe frontal sont bien alimentes

Table 6.6: Examples of synthetically generated French sentences using different augmentation techniques

it by generating a new sentence using each of these methods to more than double the original size. Upon training our DA classifier on this dataset with number of epochs=10, we obtain the results highlighted in Table 6.5.

The accuracy of our BERT-based DA classifier increases by 4% and jumps to a solid 72% when trained on data augmented using Random Insertion technique. Similar result is also obtained upon training the classifier on GPT-2 augmented data. A modest improvement in accuracy is also observed for data augmentation techniques like Synonym Replacement, Back Translation, and Random Delete. Model performance however dips by 1% upon using EDA augmented data during training. Main reason for this might be because of the poor quality of the newly generated data. Similar results are also observed in our m-BERT based DA classifier. When trained on data augmented using Random Insertion, the model accuracy increases by 4% and becomes as high as 84% which is quite impressive. Slight improvement in accuracy is also observed for augmentation techniques like Back Translation and Synonym Replacement. All in all, although most of the data augmentation techniques were able to boost the performance of our DA classifiers on French dataset, low quality of the generated data may have prevented it from achieving exceptional results. To validate our claim, we take a look at some of the sentences that were newly generated. Table 6.6 shows how a French sentence with the tag 'Yes/No Question' was manipulated to generate 6 new synthetic examples. However, some of them no longer preserve the same label (e.g., the sentence using Back Translation is not a Yes/No Question anymore but a Statement). Such mislabelling in augmented training data may hinder the model in properly learning the accurate class features during training. Future work might look into ways to identify and remove newly generated sentences that no longer preserve the original class label for better model performance.

Chapter 7

Conclusions, Recommendations, & Future Work

Dialog systems have gained traction in the recent years by showcasing a great promise in interacting with humans using natural language text. Classifying the intent of a user dialog in a conversation, also known as dialog act, is a key component in building these conversational agents. By identifying the different dialog acts, chatbots can respond more coherently and assist users in accomplishing their tasks more effectively. In this work, we have addressed the problem of recognizing user dialog acts by open domain dialog systems. We have introduced a fine-tuned pretrained BERT-based dialog act classifier applicable for both of our conversational agents, ANA and MIRA. For this, we first investigated the current literature and through iterative discussions, proposed a taxonomy of 8 dialog acts that are suitable to capture the intents of our chatbot users. We then curated a high-quality, large-scale dataset consisting of ∼24k user utterances from a wide range of domains like mental health, airlines, banking, product reviews, insurance, movie reviews and so on. Upon fine-tuning our proposed classifier on this dataset, it outperformed the baseline SVM model by achieving SOTA accuracy. Through further evaluations, we prove the generalizability and robustness of our proposed model on unseen dataset. Given how difficult it is to curate adequate labelled dataset for domain specific DAs, we look into the feasibility of implementing a wide range of data augmentation techniques to augment the existing training data and improve model performance. We first provide a brief overview of the different augmentation techniques that are out there for text data by categorizing them into an easy to understand taxonomy. Next, we compare the effectiveness of some of these methods by implementing them on a number of NLP tasks like news classification, sentiment analysis, emotion detection, slot filling and intent detection. Next, we apply the knowledge gained from these experiments into improving the performance of our DA classifier in low resource setting. Through extensive experiments, we show that, in a simulated low data regime with only 10 examples per label, methods as simple as synonym replacement can double the size of training data and improve the performance of our DA classifier by ∼8%. Lastly, in the direction of building multilingual conversational agents, we demonstrated how our proposed classifier and augmentation techniques can be adapted to effectively detect dialog acts from French utterances.

As for future work, we suggest investigating the following avenues:

i Although we had structured dialog act recognition as a multi-class classification problem, there can be instances where a single sentence is used to express multiple dialog acts. For example, 'I wonder where he's going' can be used to both convey a Statement and a Question. Structuring dialog act recognition as a multi-label classification problem to handle such instances would be interesting.

ii When detecting user dialog acts, often times, the previous utterances can provide important context. For example, 'So she can't go' can be tagged as either a Question or a Statement depending on the previous dialogs in the conversation. Although our proposed dataset does not include complete user conversations, this is definitely something worth exploring.

iii In our work, we had decided to use 8 dialog acts to capture our user intents. However, it might be necessary to further distinguish between some of them. For example, an open-ended question from a user can be either Factual (requires

Wikipedia as a source) or Opinion-based (requires Twitter as a source). Whether the inclusion of more dialog acts will be beneficial for our DA classifier is an interesting topic to investigate.

iv Lastly, on the topic of text data augmentation, it would be worth exploring its effectiveness for improving the performance of DA classifiers by augmenting training dataset curated in low resource languages like Urdu, Bengali, and Punjabi.

Bibliography

[17] J. Valsiner and A. U. Branco, *Communication and metacommunication in human development*. IAP, 2006.

[18] R. Genç, "The importance of communication in sustainability & sustainable strategies," *Procedia Manufacturing*, vol. 8, pp. 511–516, 2017.

[19] M McTear, Z Callejas, and D Griol, "The conversational interface: Talking to smart devices: Springer international publishing," *Doi: https://doi.org/10.1007/978-3-319-32967-3*, 2016.

[20] W. Ingber, "Linguistic communication and speech acts," *The Philosophical Review*, vol. 91, no. 1, pp. 134–138, 1982.

[21] *Speech acts (stanford encyclopedia of philosophy)*, https://plato.stanford.edu/entries/speech-acts/, (Accessed on 01/05/2023).

[22] C. Wei, Z. Yu, and S. Fong, "How to build a chatbot: Chatbot framework and its capabilities," in *Proceedings of the 2018 10th international conference on machine learning and computing*, 2018, pp. 369–373.

[23] A. Xu, Z. Liu, Y. Guo, V. Sinha, and R. Akkiraju, "A new chatbot for customer service on social media," in *Proceedings of the 2017 CHI conference on human factors in computing systems*, 2017, pp. 3506–3510.

[24] J. J. Godfrey and E. Holliman, "Switchboard-1 release 2," *Linguistic Data Consortium, Philadelphia*, vol. 926, p. 927, 1997.

[25] R. Dhillon, S. Bhagat, H. Carvey, and E. Shriberg, "Meeting recorder project: Dialog act labeling guide," INTERNATIONAL COMPUTER SCIENCE INST BERKELEY CA, Tech. Rep., 2004.

[26] P. Colombo, E. Chapuis, M. Manica, E. Vignon, G. Varni, and C. Clavel, "Guiding attention in sequence-to-sequence models for dialogue act prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 7594–7601.

[27] R. Li, C. Lin, M. Collinson, X. Li, and G. Chen, "A dual-attention hierarchical recurrent neural network for dialogue act classification," *arXiv preprint arXiv:1810.09154*, 2018.

[28] G. Malhotra, A. Waheed, A. Srivastava, M. S. Akhtar, and T. Chakraborty, "Speaker and time-aware joint contextual learning for dialogue-act classification in counselling conversations," in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 2022, pp. 735–745.

[29] D. Gautam, N. Maharjan, A. C. Graesser, and V. Rus, "Automated speech act categorization of chat utterances in virtual internships," in *EDM*, 2018.

[30] K. Quinn and O. Zaiane, "Identifying questions & requests in conversation," in *Proceedings of the 2014 International C* Conference on Computer Science & Software Engineering*, 2014, pp. 1–6.

[31] A. Adadi, "A survey on data-efficient algorithms in big data era," *Journal of Big Data*, vol. 8, no. 1, pp. 1–54, 2021.

[32] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.

[33] J. Wei, C. Huang, S. Vosoughi, Y. Cheng, and S. Xu, "Few-shot text classification with triplet networks, data augmentation, and curriculum learning," *arXiv preprint arXiv:2103.07552*, 2021.

[34] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," *arXiv preprint arXiv:1511.06709*, 2015.

[35] V. Kumar, A. Choudhary, and E. Cho, "Data augmentation using pre-trained transformer models," *arXiv preprint arXiv:2003.02245*, 2020.

[36] J. Wei and K. Zou, "Eda: Easy data augmentation techniques for boosting performance on text classification tasks," *arXiv preprint arXiv:1901.11196*, 2019.

[37] H. Zhu, L. Dong, F. Wei, B. Qin, and T. Liu, "Transforming wikipedia into augmented data for query-focused summarization," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2022.

[38] A. Asai and H. Hajishirzi, "Logic-guided data augmentation and regularization for consistent question answering," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online: Association for Computational Linguistics, Jul. 2020, pp. 5642–5650. DOI: 10.18653/v1/2020.acl-main.499. [Online]. Available: https://aclanthology.org/2020.acl-main.499.

[39] C.-M. Lai, M.-H. Hsu, C.-W. Huang, and Y.-N. Chen, "Controllable user dialogue act augmentation for dialogue state tracking," *arXiv preprint arXiv:2207.12757*, 2022.

[40] A. Welivita and P. Pu, "A taxonomy of empathetic response intents in human social conversations," in *Proceedings of the 28th International Conference on Computational Linguistics*, Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 4886–4899. DOI: 10.18653/v1/2020.coling-main.429. [Online]. Available: https://aclanthology.org/2020.coling-main.429.

[41] G. Leech and M. Weisser, "Generic speech act annotation for task-oriented dialogues," in *Proceedings of the corpus linguistics 2003 conference*, Lancaster: Lancaster University, vol. 16, 2003, pp. 441–446.

[42] C. T. Hemphill, J. J. Godfrey, and G. R. Doddington, "The ATIS spoken language systems pilot corpus," in *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*, 1990. [Online]. Available: https://aclanthology.org/H90-1021.

[43] D. Jurafsky *et al.*, "Automatic detection of discourse structure for speech recognition and understanding," in *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, IEEE, 1997, pp. 88–95.

[44] P. Budzianowski *et al.*, "MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 5016–5026. DOI: 10.18653/v1/D18-1547. [Online]. Available: https://aclanthology.org/D18-1547.

[45] Y. Li, H. Su, X. Shen, W. Li, Z. Cao, and S. Niu, "DailyDialog: A manually labelled multi-turn dialogue dataset," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Taipei, Taiwan: Asian Federation of Natural Language Processing, Nov. 2017, pp. 986–995. [Online]. Available: https://aclanthology.org/I17-1099.

[46] D. Amanova, V. Petukhova, and D. Klakow, "Creating annotated dialogue resources: Cross-domain dialogue act classification," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, Portorož, Slovenia: European Language Resources Association (ELRA), May 2016, pp. 111–117. [Online]. Available: https://aclanthology.org/L16-1017.

[47] W. Peng, Y. Hu, L. Xing, Y. Xie, X. Zhang, and Y. Sun, "Modeling intention, emotion and external world in dialogue systems," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2022, pp. 7042–7046.

[48] S. Paul, R. Goel, and D. Hakkani-Tür, "Towards universal dialogue act tagging for task-oriented dialogues," *arXiv preprint arXiv:1907.03020*, 2019.

[49] G. Bilquise, S. Ibrahim, K. Shaalan, *et al.*, "Emotionally intelligent chatbots: A systematic literature review," *Human Behavior and Emerging Technologies*, vol. 2022, 2022.

[50] T. Saha, A. Patra, S. Saha, and P. Bhattacharyya, "Towards emotion-aided multi-modal dialogue act classification," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online: Association for Computational Linguistics, Jul. 2020, pp. 4361–4372. DOI: 10.18653/v1/2020.acl-main.402. [Online]. Available: https://aclanthology.org/2020.acl-main.402.

[51] S. Poria, D. Hazarika, N. Majumder, G. Naik, E. Cambria, and R. Mihalcea, “Meld: A multimodal multi-party dataset for emotion recognition in conversations,” *arXiv preprint arXiv:1810.02508*, 2018.

[52] C. Busso *et al.*, “Iemocap: Interactive emotional dyadic motion capture database,” *Language resources and evaluation*, vol. 42, pp. 335–359, 2008.

[53] B. Samei, H. Li, F. Keshtkar, V. Rus, and A. C. Graesser, “Context-based speech act classification in intelligent tutoring systems,” in *International conference on intelligent tutoring systems*, Springer, 2014, pp. 236–241.

[54] J. Arguello and K. Shaffer, “Predicting speech acts in mooc forum posts,” in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 9, 2015, pp. 2–11.

[55] W. Kraaij, T. Hain, M. Lincoln, and W. Post, “The ami meeting corpus,” 2005.

[56] W. W. Cohen, V. R. Carvalho, and T. M. Mitchell, “Learning to classify email into “speech acts”,” in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 309–316. [Online]. Available: https://aclanthology.org/W04-3240.

[57] V. Embar, *Leveraging speech acts for conversational ai*, https://blog.webex.com/engineering/leveraging-speech-acts-for-conversational-ai/, (Accessed on 11/24/2022), 2021.

[58] T. Saha, S. Saha, and P. Bhattacharyya, “Tweet act classification : A deep learning based classifier for recognizing speech acts in twitter,” in *2019 International Joint Conference on Neural Networks (IJCNN)*, 2019, pp. 1–8. DOI: 10.1109/IJCNN.2019.8851805.

[59] J. R. Searle, “A classification of illocutionary acts1,” *Language in society*, vol. 5, no. 1, pp. 1–23, 1976.

[60] R. Zhang, D. Gao, and W. Li, “What are tweeters doing: Recognizing speech acts in twitter,” in *Analyzing Microtext*, 2011.

[61] G. Penha, A. Balan, and C. Hauff, “Introducing mantis: A novel multi-domain information seeking dialogues dataset,” *arXiv preprint arXiv:1912.04639*, 2019.

[62] C. Qu, L. Yang, W. B. Croft, J. R. Trippas, Y. Zhang, and M. Qiu, “Analyzing and characterizing user intent in information-seeking conversations,” in *The 41st international acm sigir conference on research & development in information retrieval*, 2018, pp. 989–992.

[63] A. Wood, P. Rodeghero, A. Armaly, and C. McMillan, “Detecting speech act types in developer question/answer conversations during bug repair,” in *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ser. ESEC/FSE 2018, Lake Buena Vista, FL, USA: Association for Computing Machinery, 2018, 491–502, ISBN: 9781450355735. DOI: 10.1145/3236024.3236031. [Online]. Available: https://doi.org/10.1145/3236024.3236031.

[64] D. Yu and Z. Yu, “Midas: A dialog act annotation scheme for open domain human machine spoken conversations,” *arXiv preprint arXiv:1908.10023*, 2019.

[65] J. M. Noble *et al.*, “Developing, implementing, and evaluating an artificial intelligence–guided mental health resource navigation chatbot for health care workers and their families during and following the covid-19 pandemic: Protocol for a cross-sectional study,” *JMIR Research Protocols*, vol. 11, no. 7, e33717, 2022.

[66] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, “Learning word vectors for sentiment analysis,” in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, Oregon, USA: Association for Computational Linguistics, Jun. 2011, pp. 142–150. [Online]. Available: https://aclanthology.org/P11-1015.

[67] S. A. Abdul-Kader, B. N. A.-d. Abed, S. A. Noaman, and J. Woods, “The extracting actionable imperative sentences from the internet for chatbot,” *SCIENCE AND WORLD*, p. 36, 2013.

[68] F. Mao, R. E. Mercer, and L. Xiao, “Extracting imperatives from wikipedia article for deletion discussions,” in *Proceedings of the first workshop on Argumentation Mining*, 2014, pp. 106–107.

[69] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, “Squad: 100,000+ questions for machine comprehension of text,” *arXiv preprint arXiv:1606.05250*, 2016.

[70] X. Li and D. Roth, “Learning question classifiers,” in *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002. [Online]. Available: https://www.aclweb.org/anthology/C02-1150.

[71] C. Clark, K. Lee, M.-W. Chang, T. Kwiatkowski, M. Collins, and K. Toutanova, “Boolq: Exploring the surprising difficulty of natural yes/no questions,” *arXiv preprint arXiv:1905.10044*, 2019.

[72] A. Coucke *et al.*, “Snips voice platform: An embedded spoken language understanding system for private-by-design voice interfaces,” *arXiv preprint arXiv:1805.10190*, 2018.

[73] B. Byrne *et al.*, “Taskmaster-1: Toward a realistic and diverse dialog dataset,” *arXiv preprint arXiv:1909.05358*, 2019.

[74] S. Acharya and G. Fung, "Using optimal embeddings to learn new intents with few examples: An application in the insurance domain," 2020. [Online]. Available: http://ceur-ws.org/Vol-2666/KDD_Converse20_paper_10.pdf.

[75] V. Raheja and J. R. Tetreault, "Dialogue act classification with context-aware self-attention," *ArXiv*, vol. abs/1904.02594, 2019.

[76] A. Ahmadvand, J. I. Choi, and E. Agichtein, "Contextual dialogue act classification for open-domain conversational agents," in *Proceedings of the 42nd international acm sigir conference on research and development in information retrieval*, 2019, pp. 1273–1276.

[77] T. Saha, S. R. Jayashree, S. Saha, and P. Bhattacharyya, "Bert-caps: A transformer-based capsule network for tweet act classification," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 5, pp. 1168–1179, 2020.

[78] T.-W. Wu, R. Su, and B.-H. Juang, "A context-aware hierarchical bert fusion network for multi-turn dialog act detection," *arXiv preprint arXiv:2109.01267*, 2021.

[79] X. Li, Y. Wang, S. Sun, S. Panda, J. Liu, and J. Gao, "Microsoft dialogue challenge: Building end-to-end task-completion dialogue systems," *arXiv preprint arXiv:1807.11125*, 2018.

[80] A. Rastogi, X. Zang, S. Sunkara, R. Gupta, and P. Khaitan, "Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 8689–8696.

[81] T. Saha, D. Gupta, S. Saha, and P. Bhattacharyya, "Emotion aided dialogue act classification for task-independent conversations in a multi-modal framework," *Cognitive Computation*, vol. 13, no. 2, pp. 277–289, 2021.

[82] L. Qin, W. Che, Y. Li, M. Ni, and T. Liu, "Dcr-net: A deep co-interactive relation network for joint dialog act recognition and sentiment classification," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, 2020, pp. 8665–8672.

[83] C. Cerisara, S. Jafaritazehjani, A. Oluokun, and H. T. Le, "Multi-task dialog act and sentiment recognition on mastodon," in *Proceedings of the 27th International Conference on Computational Linguistics*, Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 745–754. [Online]. Available: https://aclanthology.org/C18-1063.

[84] B. Huber *et al.*, "Specialtime: Automatically detecting dialogue acts from speech to support parent-child interaction therapy," in *Proceedings of the 13th EAI International Conference on Pervasive Computing Technologies for Healthcare*, 2019, pp. 139–148.

[85] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[86] C. Lou, *Artificial neural networks: Their training process and applications*, https://www.whitman.edu/documents/Academics/Mathematics/2019/Lou-Hundley.pdf, (Accessed on 01/16/2023).

[87] H. Robbins and S. Monro, “A stochastic approximation method,” *The annals of mathematical statistics*, pp. 400–407, 1951.

[88] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[89] D. Jurafsky, *Speech & language processing.* Pearson Education India, 2000.

[90] D. Jurafsky and J. H. Martin, “Speech and language processing. vol. 3,” *US: Prentice Hall*, 2014.

[91] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning internal representations by error propagation,” California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.

[92] S. Antol *et al.*, “Vqa: Visual question answering,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2425–2433.

[93] G. Erkan and D. R. Radev, “Lexrank: Graph-based lexical centrality as salience in text summarization,” *Journal of artificial intelligence research*, vol. 22, pp. 457–479, 2004.

[94] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan, “A diversity-promoting objective function for neural conversation models,” *arXiv preprint arXiv:1510.03055*, 2015.

[95] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, “Deep learning–based text classification: A comprehensive review,” *ACM Computing Surveys (CSUR)*, vol. 54, no. 3, pp. 1–40, 2021.

[96] T. Young, D. Hazarika, S. Poria, and E. Cambria, “Recent trends in deep learning based natural language processing,” *ieee Computational intelligenCe magazine*, vol. 13, no. 3, pp. 55–75, 2018.

[97] J. Cahn, “Chatbot: Architecture, design, & development,” *University of Pennsylvania School of Engineering and Applied Science Department of Computer and Information Science*, 2017.

[98] K. Cho *et al.*, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.

[99] K. Ramesh, S. Ravishankaran, A. Joshi, and K Chandrasekaran, “A survey of design techniques for conversational agents,” in *International conference on information, communication and computing technology*, Springer, 2017, pp. 336–350.

[100] H. Sak, A. W. Senior, and F. Beaufays, “Long short-term memory recurrent neural network architectures for large scale acoustic modeling,” 2014.

[101] A. Vaswani *et al.*, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.

[102] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[103] D. Lukovnikov, A. Fischer, and J. Lehmann, "Pretrained transformers for simple question answering over knowledge graphs," in *International Semantic Web Conference*, Springer, 2019, pp. 470–486.

[104] Q. Wang *et al.*, "Learning deep transformer models for machine translation," *arXiv preprint arXiv:1906.01787*, 2019.

[105] *5 types of classification algorithms in machine learning*, https://monkeylearn.com/blog/classification-algorithms/, (Accessed on 01/18/2023).

[106] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*, 1992, pp. 144–152.

[107] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine learning*, vol. 46, no. 1, pp. 389–422, 2002.

[108] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[109] V. Vapnik, S. Golowich, and A. Smola, "Support vector method for function approximation, regression estimation and signal processing," *Advances in neural information processing systems*, vol. 9, 1996.

[110] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *European conference on machine learning*, Springer, 1998, pp. 137–142.

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

[112] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. [Online]. Available: https://aclanthology.org/N19-1423.

[113] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[114] Y. Chen, Y. Liu, L. Chen, and Y. Zhang, "DialogSum: A real-life scenario dialogue summarization dataset," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, Online: Association for Computational Linguistics, Aug. 2021, pp. 5062–5074. DOI: 10.18653/v1/2021.findings-acl.449. [Online]. Available: https://aclanthology.org/2021.findings-acl.449.

[115] K. Sun, D. Yu, J. Chen, D. Yu, Y. Choi, and C. Cardie, "DREAM: A challenge data set and models for dialogue-based reading comprehension," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 217–231, 2019. DOI: 10.1162/tacl_a_00264. [Online]. Available: https://aclanthology.org/Q19-1014.

[116] L. Cui, Y. Wu, S. Liu, Y. Zhang, and M. Zhou, "MuTual: A dataset for multi-turn dialogue reasoning," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online: Association for Computational Linguistics, Jul. 2020, pp. 1406–1416. DOI: 10.18653/v1/2020.acl-main.130. [Online]. Available: https://aclanthology.org/2020.acl-main.130.

[117] S. Yang, W. Xiao, M. Zhang, S. Guo, J. Zhao, and F. Shen, "Image data augmentation for deep learning: A survey," *arXiv preprint arXiv:2204.08610*, 2022.

[118] C. Coulombe, "Text data augmentation made simple by leveraging nlp cloud apis," *arXiv preprint arXiv:1812.04718*, 2018.

[119] P. Liu, X. Wang, C. Xiang, and W. Meng, "A survey of text data augmentation," in *2020 International Conference on Computer Communication and Network Security (CCNS)*, IEEE, 2020, pp. 191–195.

[120] S. Y. Feng *et al.*, "A survey of data augmentation approaches for nlp," *arXiv preprint arXiv:2105.03075*, 2021.

[121] C. Shorten, T. M. Khoshgoftaar, and B. Furht, "Text data augmentation for deep learning," *Journal of big Data*, vol. 8, no. 1, pp. 1–34, 2021.

[122] M. Bayer, M.-A. Kaufhold, and C. Reuter, "A survey on data augmentation for text classification," *ACM Computing Surveys*, 2021.

[123] B. Li, Y. Hou, and W. Che, "Data augmentation approaches in natural language processing: A survey," *AI Open*, 2022.

[124] J. Chen, D. Tam, C. Raffel, M. Bansal, and D. Yang, "An empirical survey of data augmentation for limited data learning in nlp," *arXiv preprint arXiv:2106.07499*, 2021.

[125] O. Kolomiyets, S. Bethard, and M.-F. Moens, "Model-portability experiments for textual temporal analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, Oregon, USA: Association for Computational Linguistics, Jun. 2011, pp. 271–276. [Online]. Available: https://aclanthology.org/P11-2047.

[126] Y. Li, T. Cohn, and T. Baldwin, "Robust training under linguistic adversity," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 21–27. [Online]. Available: https://aclanthology.org/E17-2004.

[127] M. Jungiewicz and A. Smywiński-Pohl, "Towards textual data augmentation for neural networks: Synonyms and maximum loss," *Computer Science*, vol. 20, 2019.

[128] A. Mosolova, V. Fomin, and I. Bondarenko, "Text augmentation for neural networks.," in *AIST (Supplement)*, 2018, pp. 104–109.

[129] X. Zuo, Y. Chen, K. Liu, and J. Zhao, "KnowDis: Knowledge enhanced data augmentation for event causality detection via distant supervision," in *Proceedings of the 28th International Conference on Computational Linguistics*, Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 1544–1550. DOI: 10.18653/v1/2020.coling-main.135. [Online]. Available: https://aclanthology.org/2020.coling-main.135.

[130] R. Xiang, E. Chersoni, Y. Long, Q. Lu, and C.-R. Huang, "Lexical data augmentation for text classification in deep learning," in *Canadian Conference on Artificial Intelligence*, Springer, 2020, pp. 521–527.

[131] V. Marivate and T. Sefara, "Improving short text classification through global augmentation methods," in *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, Springer, 2020, pp. 385–399.

[132] G. Rizos, K. Hemker, and B. Schuller, "Augment to prevent: Short-text data augmentation in deep learning for hate-speech classification," in *Proceedings of the 28th ACM international conference on information and knowledge management*, 2019, pp. 991–1000.

[133] X. Zhang and Y. LeCun, "Text understanding from scratch," *arXiv preprint arXiv:1502.01710*, 2015.

[134] W. Y. Wang and D. Yang, "That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets," in *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, pp. 2557–2563.

[135] J. Li, S. Ji, T. Du, B. Li, and T. Wang, "Textbugger: Generating adversarial text against real-world applications," *arXiv preprint arXiv:1812.05271*, 2018.

[136] K. J. Madukwe, X. Gao, and B. Xue, "Token replacement-based data augmentation methods for hate speech detection," *World Wide Web*, vol. 25, no. 3, pp. 1129–1150, 2022.

[137] N. Mrkšić *et al.*, "Counter-fitting word vectors to linguistic constraints," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 142–148. DOI: 10.18653/v1/N16-1018. [Online]. Available: https://aclanthology.org/N16-1018.

[138] O. Levy and Y. Goldberg, “Dependency-based word embeddings,” in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Baltimore, Maryland: Association for Computational Linguistics, Jun. 2014, pp. 302–308. DOI: 10.3115/v1/P14-2050. [Online]. Available: https://aclanthology.org/P14-2050.

[139] M. Alzantot, Y. Sharma, A. Elgohary, B.-J. Ho, M. Srivastava, and K.-W. Chang, “Generating natural language adversarial examples,” *arXiv preprint arXiv:1804.07998*, 2018.

[140] C. Chelba *et al.*, “One billion word benchmark for measuring progress in statistical language modeling,” *arXiv preprint arXiv:1312.3005*, 2013.

[141] F. Gao *et al.*, “Soft contextual data augmentation for neural machine translation,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 5539–5544.

[142] S. Kobayashi, “Contextual augmentation: Data augmentation by words with paradigmatic relations,” *arXiv preprint arXiv:1805.06201*, 2018.

[143] Y. Belinkov and Y. Bisk, “Synthetic and natural noise both break neural machine translation,” *arXiv preprint arXiv:1711.02173*, 2017.

[144] S. Y. Feng, V. Gangal, D. Kang, T. Mitamura, and E. Hovy, “Genaug: Data augmentation for finetuning text generators,” *arXiv preprint arXiv:2010.01794*, 2020.

[145] A. Karimi, L. Rossi, and A. Prati, “Aeda: An easier data augmentation technique for text classification,” *arXiv preprint arXiv:2108.13230*, 2021.

[146] Z. Miao, Y. Li, X. Wang, and W.-C. Tan, “Snippext: Semi-supervised opinion mining with augmented data,” in *Proceedings of The Web Conference 2020*, 2020, pp. 617–628.

[147] C. Rastogi, N. Mofid, and F.-I. Hsiao, “Can we achieve more with less? exploring data augmentation for toxic comment classification,” *arXiv preprint arXiv:2007.00875*, 2020.

[148] T. Niu and M. Bansal, “Adversarial over-sensitivity and over-stability strategies for dialogue models,” *arXiv preprint arXiv:1809.02079*, 2018.

[149] M. Artetxe, G. Labaka, E. Agirre, and K. Cho, “Unsupervised neural machine translation,” *arXiv preprint arXiv:1710.11041*, 2017.

[150] G. Lample, A. Conneau, L. Denoyer, and M. Ranzato, “Unsupervised machine translation using monolingual corpora only,” *arXiv preprint arXiv:1711.00043*, 2017.

[151] M. Regina, M. Meyer, and S. Goutal, “Text data augmentation: Towards better detection of spear-phishing emails,” *arXiv preprint arXiv:2007.02033*, 2020.

[152] G. Yan, Y. Li, S. Zhang, and Z. Chen, “Data augmentation for deep learning of judgment documents,” in *International Conference on Intelligent Science and Big Data Engineering*, Springer, 2019, pp. 232–242.

[153] S. Yu, J. Yang, D. Liu, R. Li, Y. Zhang, and S. Zhao, "Hierarchical data augmentation and the application in text classification," *IEEE Access*, vol. 7, pp. 185 476–185 485, 2019.

[154] S. Edunov, M. Ott, M. Auli, and D. Grangier, "Understanding back-translation at scale," *arXiv preprint arXiv:1808.09381*, 2018.

[155] L. Marceau, R. Belbahar, M. Queudot, E. Charton, and M.-J. Meurs, "Quick starting dialog systems with paraphrase generation," *arXiv preprint arXiv:2204.02546*, 2022.

[156] Q. Xie, Z. Dai, E. Hovy, T. Luong, and Q. Le, "Unsupervised data augmentation for consistency training," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6256–6268, 2020.

[157] M. Iyyer, J. Wieting, K. Gimpel, and L. Zettlemoyer, "Adversarial example generation with syntactically controlled paraphrase networks," *arXiv preprint arXiv:1804.06059*, 2018.

[158] M. Chen, Q. Tang, S. Wiseman, and K. Gimpel, "Controllable paraphrase generation with a syntactic exemplar," *arXiv preprint arXiv:1906.00565*, 2019.

[159] A. Gupta, A. Agarwal, P. Singh, and P. Rai, "A deep generative framework for paraphrase generation," in *Proceedings of the aaai conference on artificial intelligence*, vol. 32, 2018.

[160] M. Bornea, L. Pan, S. Rosenthal, R. Florian, and A. Sil, "Multilingual transfer learning for qa using translation as data augmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 2021, pp. 12 583–12 591.

[161] V. Barriere and A. Balahur, "Improving sentiment analysis over non-English tweets using multilingual transformers and automatic translation for data-augmentation," in *Proceedings of the 28th International Conference on Computational Linguistics*, Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 266–271. DOI: 10.18653/v1/2020.coling-main.23. [Online]. Available: https://aclanthology.org/2020.coling-main.23.

[162] A. Perevalov and A. Both, "Augmentation-based answer type classification of the smart dataset.," in *SMART@ ISWC*, 2020, pp. 1–9.

[163] D. Liu *et al.*, "Tell me how to ask again: Question data augmentation with controllable rewriting in continuous space," *arXiv preprint arXiv:2010.01475*, 2020.

[164] Y. Hou, Y. Liu, W. Che, and T. Liu, "Sequence-to-sequence data augmentation for dialogue language understanding," *arXiv preprint arXiv:1807.01554*, 2018.

[165] S. Wang, R. Gupta, N. Chang, and J. Baldridge, "A task in a suit and a tie: Paraphrase generation with semantic augmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 7176–7183.

[166] N. Malandrakis, M. Shen, A. Goyal, S. Gao, A. Sethi, and A. Metallinou, "Controlled text generation for data augmentation in intelligent artificial agents," in *Proceedings of the 3rd Workshop on Neural Generation and Translation*, Hong Kong: Association for Computational Linguistics, Nov. 2019, pp. 90–98. DOI: 10.18653/v1/D19-5609. [Online]. Available: https://aclanthology.org/D19-5609.

[167] Y. Cao and X. Wan, "DivGAN: Towards diverse paraphrase generation via diversified generative adversarial network," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, Online: Association for Computational Linguistics, Nov. 2020, pp. 2411–2421. DOI: 10.18653/v1/2020.findings-emnlp.218. [Online]. Available: https://aclanthology.org/2020.findings-emnlp.218.

[168] J. Xu, X. Ren, J. Lin, and X. Sun, "Dp-gan: Diversity-promoting generative adversarial network for generating informative and diversified text," *arXiv preprint arXiv:1802.01345*, 2018.

[169] M. T. Ribeiro, S. Singh, and C. Guestrin, "Semantically equivalent adversarial rules for debugging NLP models," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 856–865. DOI: 10.18653/v1/P18-1079. [Online]. Available: https://aclanthology.org/P18-1079.

[170] X. Wu, S. Lv, L. Zang, J. Han, and S. Hu, "Conditional bert contextual augmentation," in *International conference on computational science*, Springer, 2019, pp. 84–95.

[171] L. Shi, D. Liu, G. Liu, and K. Meng, "Aug-bert: An efficient data augmentation algorithm for text classification," in *International Conference in Communications, Signal Processing, and Systems*, Springer, 2019, pp. 2191–2198.

[172] S. Garg and G. Ramakrishnan, "BAE: BERT-based adversarial examples for text classification," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online: Association for Computational Linguistics, Nov. 2020, pp. 6174–6181. DOI: 10.18653/v1/2020.emnlp-main.498. [Online]. Available: https://aclanthology.org/2020.emnlp-main.498.

[173] K. Pantelidou, D. Chatzakou, T. Tsikrika, S. Vrochidis, and I. Kompatsiaris, "Selective word substitution for contextualized data augmentation," in *International Conference on Applications of Natural Language to Information Systems*, Springer, 2022, pp. 508–516.

[174] Q. Yu and X. Zhang, "Application of data augmentation in financial sentiment analysis task," *Available at SSRN 3969563*, 2021.

[175] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[176] Y. Liu *et al.*, “Roberta: A robustly optimized bert pretraining approach,” *arXiv preprint arXiv:1907.11692*, 2019.

[177] A. Anaby-Tavor *et al.*, “Do not have enough data? deep learning to the rescue!” In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 7383–7390.

[178] H. Quteineh, S. Samothrakis, and R. Sutcliffe, “Textual data augmentation for efficient active learning on tiny datasets,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online: Association for Computational Linguistics, Nov. 2020, pp. 7400–7410. DOI: 10.18653/v1/2020.emnlp-main.600. [Online]. Available: https://aclanthology.org/2020.emnlp-main.600.

[179] V. Claveau, A. Chaffin, and E. Kijak, “Generating artificial texts as substitution or complement of training data,” *arXiv preprint arXiv:2110.13016*, 2021.

[180] K. M. Yoo, D. Park, J. Kang, S.-W. Lee, and W. Park, “Gpt3mix: Leveraging large-scale language models for text augmentation,” *arXiv preprint arXiv:2104.08826*, 2021.

[181] U. Azam, H. Rizwan, and A. Karim, “Exploring data augmentation strategies for hate speech detection in roman urdu,” in *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, 2022, pp. 4523–4531.

[182] T. Brown *et al.*, “Language models are few-shot learners,” *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[183] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio, “Generating sentences from a continuous space,” *arXiv preprint arXiv:1511.06349*, 2015.

[184] Z. Hu, Z. Yang, X. Liang, R. Salakhutdinov, and E. P. Xing, “Toward controlled generation of text,” in *International conference on machine learning*, PMLR, 2017, pp. 1587–1596.

[185] K. Guu, T. B. Hashimoto, Y. Oren, and P. Liang, “Generating sentences by editing prototypes,” *Transactions of the Association for Computational Linguistics*, vol. 6, pp. 437–450, 2018.

[186] G. Russo, N. Hollenstein, C. Musat, and C. Zhang, “Control, generate, augment: A scalable framework for multi-attribute text generation,” *arXiv preprint arXiv:2004.14983*, 2020.

[187] F. Piedboeuf and P. Langlais, “Effective data augmentation for sentence classification using one vae per class,” in *Proceedings of the 29th International Conference on Computational Linguistics*, 2022, pp. 3454–3464.

[188] W. Nie, N. Narodytska, and A. Patel, “Relgan: Relational generative adversarial networks for text generation,” in *International conference on learning representations*, 2018.

[189] S. Yean, P. Somani, B. S. Lee, and H. L. Oh, "Gan+: Data augmentation method using generative adversarial networks and dirichlet for indoor localisation," 2021.

[190] H. Wang, Z. Qin, and T. Wan, "Text generation based on generative adversarial nets with latent variables," in *Pacific-Asia conference on knowledge discovery and data mining*, Springer, 2018, pp. 92–103.

[191] S. Shehnepoor, R. Togneri, W. Liu, and M. Bennamoun, "Scoregan: A fraud review detector based on regulated gan with data augmentation," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 280–291, 2021.

[192] G. G. Şahin and M. Steedman, "Data augmentation via dependency tree morphing for low-resource languages," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 5004–5009. DOI: 10.18653/v1/D18-1545. [Online]. Available: https://aclanthology.org/D18-1545.

[193] S. Louvan and B. Magnini, "Simple is better! lightweight data augmentation for low resource slot filling and intent classification," in *Proceedings of the 34th Pacific Asia Conference on Language, Information and Computation*, Hanoi, Vietnam: Association for Computational Linguistics, Oct. 2020, pp. 167–177. [Online]. Available: https://aclanthology.org/2020.paclic-1.20.

[194] J. Min, R. T. McCoy, D. Das, E. Pitler, and T. Linzen, "Syntactic data augmentation increases robustness to inference heuristics," *arXiv preprint arXiv:2004.11999*, 2020.

[195] J. F. Steffensen, *Interpolation.* Courier Corporation, 2006.

[196] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[197] H. Guo, Y. Mao, and R. Zhang, "Augmenting data with mixup for sentence classification: An empirical study," *arXiv preprint arXiv:1905.08941*, 2019.

[198] S. Yoon, G. Kim, and K. Park, "Ssmix: Saliency-based span mixup for text classification," *arXiv preprint arXiv:2106.08062*, 2021.

[199] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[200] A. Fernández, S. Garcia, F. Herrera, and N. V. Chawla, "Smote for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary," *Journal of artificial intelligence research*, vol. 61, pp. 863–905, 2018.

[201] N. Curukoglu and A. Ozpinar, "Smote-text: A modified smote for turkish text classification," in *The International Conference on Artificial Intelligence and Applied Mathematics in Engineering*, Springer, 2020, pp. 82–92.

[202] J. Wang, W. F. Lu, and H. T. Loh, "P-smote: One oversampling technique for class imbalanced text classification," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 54792, 2011, pp. 1089–1098.

[203] V. Rupapara, F. Rustam, H. F. Shahzad, A. Mehmood, I. Ashraf, and G. S. Choi, "Impact of smote on imbalanced text features for toxic comments classification using rvvc model," *IEEE Access*, vol. 9, pp. 78 621–78 634, 2021.

[204] P. Sarakit, T. Theeramunkong, and C. Haruechaiyasak, "Improving emotion classification in imbalanced youtube dataset using smote algorithm," in *2015 2nd International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA)*, IEEE, 2015, pp. 1–5.

[205] M. Umer *et al.*, "Scientific papers citation analysis using textual features and smote resampling techniques," *Pattern Recognition Letters*, vol. 150, pp. 250–257, 2021.

[206] S. Ren, J. Zhang, L. Li, X. Sun, and J. Zhou, "Text autoaugment: Learning compositional augmentation policy for text classification," *arXiv preprint arXiv:2109.00523*, 2021.

[207] J. Wei, C. Huang, S. Xu, and S. Vosoughi, "Text augmentation in a multi-task view," *arXiv preprint arXiv:2101.05469*, 2021.

[208] R. Liu, G. Xu, C. Jia, W. Ma, L. Wang, and S. Vosoughi, "Data boost: Text data augmentation through reinforcement learning guided conditional generation," *arXiv preprint arXiv:2012.02952*, 2020.

[209] J. Min, R. T. McCoy, D. Das, E. Pitler, and T. Linzen, "Syntactic data augmentation increases robustness to inference heuristics," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online: Association for Computational Linguistics, Jul. 2020, pp. 2339–2352. DOI: 10.18653/v1/2020.acl-main.212. [Online]. Available: https://aclanthology.org/2020.acl-main.212.

[210] J. Singh, B. McCann, N. S. Keskar, C. Xiong, and R. Socher, "Xlda: Cross-lingual data augmentation for natural language inference and question answering," *ArXiv*, vol. abs/1905.11471, 2019.

[211] M. H. Shakeel, A. Karim, and I. Khan, "A multi-cascaded model with data augmentation for enhanced paraphrase detection in short texts," *Information Processing & Management*, vol. 57, no. 3, p. 102 204, 2020.

[212] R. T. Anchiêta, R. F. d. Sousa, and T. A. Pardo, "Modeling the paraphrase detection task over a heterogeneous graph network with data augmentation," *Information*, vol. 11, no. 9, p. 422, 2020.

[213] A. Fadel, I. Tuffaha, and M. Al-Ayyoub, "Tha3aroon at nsurl-2019 task 8: Semantic question similarity in arabic," *arXiv preprint arXiv:1912.12514*, 2019.

[214] Y. Wang, F. Liu, K. Verspoor, and T. Baldwin, "Evaluating the utility of model configurations and data augmentation on clinical semantic textual similarity," in *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*, Online: Association for Computational Linguistics, Jul. 2020, pp. 105–111. DOI: 10.18653/v1/2020.bionlp-1.11. [Online]. Available: https://aclanthology.org/2020.bionlp-1.11.

[215] S. Xu, J. Zhang, J. Chen, and L. Qin, "Erroneous data generation for grammatical error correction," in *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, 2019, pp. 149–158.

[216] Z. Wan, X. Wan, and W. Wang, "Improving grammatical error correction with data augmentation by editing latent representation," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 2202–2212.

[217] M. White and A. Rozovskaya, "A comparative study of synthetic data generation methods for grammatical error correction," in *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, 2020, pp. 198–208.

[218] X.-P. Nguyen, S. Joty, K. Wu, and A. T. Aw, "Data diversification: A simple strategy for neural machine translation," *Advances in Neural Information Processing Systems*, vol. 33, pp. 10 018–10 029, 2020.

[219] W. Peng, C. Huang, T. Li, Y. Chen, and Q. Liu, "Dictionary-based data augmentation for cross-domain neural machine translation," *arXiv preprint arXiv:2004.02577*, 2020.

[220] D. Li, I Te, N. Arivazhagan, C. Cherry, and D. Padfield, "Sentence boundary augmentation for neural machine translation robustness," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2021, pp. 7553–7557.

[221] S. Parida and P. Motlicek, "Abstract text summarization: A low resource challenge," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 5994–5998.

[222] A. R. Fabbri *et al.*, "Improving zero and few-shot abstractive summarization with intermediate fine-tuning and data augmentation," *arXiv preprint arXiv:2010.12836*, 2020.

[223] W. Yang, Y. Xie, L. Tan, K. Xiong, M. Li, and J. Lin, "Data augmentation for bert fine-tuning in open-domain question answering," *arXiv preprint arXiv:1904.06652*, 2019.

[224] A. Riabi, T. Scialom, R. Keraron, B. Sagot, D. Seddah, and J. Staiano, "Synthetic data augmentation for zero-shot cross-lingual question answering," *arXiv preprint arXiv:2010.12643*, 2020.

[225] Y. Shen, J. Li, S. Huang, Y. Zhou, X. Xie, and Q. Zhao, “Data augmentation for low-resource word segmentation and pos tagging of ancient chinese texts,” in *Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages*, 2022, pp. 169–173.

[226] W. Dongbo *et al.*, “Sikubert and sikuroberta: Research on the construction and application of pre-training model of sikuquanshu for digital humanities,” *Library Tribune*, pp. 1–14, 2021.

[227] C. Vania, Y. Kementchedjhieva, A. Søgaard, and A. Lopez, “A systematic comparison of methods for low-resource dependency parsing on genuinely low-resource languages,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 1105–1116. DOI: 10.18653/v1/D19-1102. [Online]. Available: https://aclanthology.org/D19-1102.

[228] K. Gulordava, P. Bojanowski, E. Grave, T. Linzen, and M. Baroni, “Colorless green recurrent networks dream hierarchically,” *arXiv preprint arXiv:1803.11138*, 2018.

[229] Z. Chen and T. Qian, “Description and demonstration guided data augmentation for sequence tagging,” *World Wide Web*, vol. 25, no. 1, pp. 175–194, 2022.

[230] S. Gao, Y. Zhang, Z. Ou, and Z. Yu, “Paraphrase augmented task-oriented dialog generation,” *arXiv preprint arXiv:2004.07462*, 2020.

[231] X. Xu, G. Wang, Y.-B. Kim, and S. Lee, “AugNLG: Few-shot natural language generation using self-trained data augmentation,” in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Online: Association for Computational Linguistics, Aug. 2021, pp. 1183–1195. DOI: 10.18653/v1/2021.acl-long.95. [Online]. Available: https://aclanthology.org/2021.acl-long.95.

[232] Y. Yin, L. Shang, X. Jiang, X. Chen, and Q. Liu, “Dialog state tracking with reinforced data augmentation,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 9474–9481.

[233] D. Chen and Z. Yu, “Gold: Improving out-of-scope detection in dialogues using data augmentation,” *ArXiv*, vol. abs/2109.03079, 2021.

[234] K. Shah, H. Patel, D. Sanghvi, and M. Shah, “A comparative analysis of logistic regression, random forest and knn models for the text classification,” *Augmented Human Research*, vol. 5, no. 1, pp. 1–16, 2020.

[235] J. Singh, G. Singh, and R. Singh, “Optimization of sentiment analysis using machine learning classifiers,” *Human-centric Computing and information Sciences*, vol. 7, no. 1, pp. 1–12, 2017.

[236] E. Saravia, H.-C. T. Liu, Y.-H. Huang, J. Wu, and Y.-S. Chen, "Carer: Contextualized affect representations for emotion recognition," in *Proceedings of the 2018 conference on empirical methods in natural language processing*, 2018, pp. 3687–3697.

[237] K. Sohn *et al.*, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," *Advances in neural information processing systems*, vol. 33, pp. 596–608, 2020.

[238] A. Kumar, K. Ahuja, R. Vadapalli, and P. Talukdar, "Syntax-guided controlled generation of paraphrases," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 329–345, 2020. DOI: 10.1162/tacl_a_00318. [Online]. Available: https://aclanthology.org/2020.tacl-1.22.

[239] Q. Chen, Z. Zhuo, and W. Wang, "Bert for joint intent classification and slot filling," *arXiv preprint arXiv:1902.10909*, 2019.

[240] P. P. Liang, C. Wu, L.-P. Morency, and R. Salakhutdinov, "Towards understanding and mitigating social biases in language models," in *International Conference on Machine Learning*, PMLR, 2021, pp. 6565–6576.

[241] C. Lee, K. Cho, and W. Kang, "Mixout: Effective regularization to finetune large-scale pretrained language models," *arXiv preprint arXiv:1909.11299*, 2019.

[242] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. arXiv: 1810.04805. [Online]. Available: http://arxiv.org/abs/1810.04805.

[243] B. Minixhofer, F. Paischer, and N. Rekabsaz, "WECHSEL: Effective initialization of subword embeddings for cross-lingual transfer of monolingual language models," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 3992–4006. [Online]. Available: https://aclanthology.org/2022.naacl-main.293.

[1] J. Weizenbaum, "Eliza—a computer program for the study of natural language communication between man and machine," *Communications of the ACM*, vol. 9, no. 1, pp. 36–45, 1966.

[2] K. M. Colby, S. Weber, and F. D. Hilf, "Artificial paranoia," *Artificial Intelligence*, vol. 2, no. 1, pp. 1–25, 1971.

[3] S. Jafarpour, C. J. Burges, and A. Ritter, "Filter, rank, and transfer the knowledge: Learning to chat," *Advances in Ranking*, vol. 10, pp. 2329–9290, 2010.

[4] A. Leuski and D. Traum, "Npceditor: Creating virtual human dialogue using information retrieval techniques," *Ai Magazine*, vol. 32, no. 2, pp. 42–56, 2011.

[5] R. Yan, Y. Song, and H. Wu, "Learning to respond with deep neural networks for retrieval-based human-computer conversation system," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2016, pp. 55–64.

[6] Z. Yan *et al.*, "Docchat: An information retrieval approach for chatbot engines using unstructured documents," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 516–525.

[7] S. Wang, D. Li, J. Geng, L. Yang, and H. Leng, "Learning to balance the coherence and diversity of response generation in generation-based chatbots," *International Journal of Advanced Robotic Systems*, vol. 17, no. 4, p. 1 729 881 420 953 006, 2020.

[8] J. Ni, T. Young, V. Pandelea, F. Xue, and E. Cambria, "Recent advances in deep learning based dialogue systems: A systematic survey," *Artificial intelligence review*, pp. 1–101, 2022.

[9] K. Hwerbi, "An ontology-based chatbot for crises management: Use case coronavirus," *arXiv preprint arXiv:2011.02340*, 2020.

[10] K. M. Colby, "Ten criticisms of parry," *ACM SIGART Bulletin*, no. 48, pp. 5–9, 1974.

[11] L. Fryer and R. Carpenter, "Bots as language learning tools," *Language Learning & Technology*, vol. 10, no. 3, pp. 8–14, 2006.

[12] E. R. Walker, R. E. McGee, and B. G. Druss, "Mortality in mental disorders and global disease burden implications: A systematic review and meta-analysis," *JAMA psychiatry*, vol. 72, no. 4, pp. 334–341, 2015.

[13] H. Dihingia, S. Ahmed, D. Borah, S. Gupta, K. Phukan, and M. K. Muchahari, "Chatbot implementation in customer service industry through deep neural networks," in *2021 International Conference on Computational Performance Evaluation (ComPE)*, IEEE, 2021, pp. 193–198.

[14] L. Bradeško and D. Mladenić, "A survey of chatbot systems through a loebner prize competition," in *Proceedings of Slovenian language technologies society eighth conference of language technologies*, Institut Jožef Stefan Ljubljana, Slovenia, 2012, pp. 34–37.

[15] R. Raine, "Making a clever intelligent agent: The theory behind the implementation," in *2009 IEEE International Conference on Intelligent Computing and Intelligent Systems*, IEEE, vol. 3, 2009, pp. 398–402.

[16] S. Singh and H. Beniwal, "A survey on near-human conversational agents," *Journal of King Saud University-Computer and Information Sciences*, 2021.

Appendix A: Chatbot Systems

Chit-chat systems are dialog systems that are designed to mimic human-behaviour. By producing natural sounding responses, chatbots converse with human beings on a wide range of events and topics, and help them accomplish multiple tasks.

A.1 Types of Chatbots

Over the years, a number of approaches have been adopted for building dialog systems.

1. Rule-Based: Such chatbots generate a response based on hand-crafted rules engineered by humans. The generated responses often sound unnatural as they do not take the contextual information in the conversation into account. A popular example of rule-based chatbot is ELIZA [1]. ELIZA takes user's utterances as input and processes it by searching for a keyword that occurs in a predefined dictionary. If the keyword is found, the utterance is mapped to a rule which then transforms the statement into a response. Otherwise, ELIZA outputs a generic response or uses an utterance from the conversation history. Few years later, another chatbot, PARRY [2], was built. Unlike ELIZA, it has an emotional state that controls the response generation process. For example, if PARRY detects anger in user input, it chooses to output a response from a predefined set of hostile responses. While these approaches may seem promising, they fail to generate an appropriate response in most of the cases. Often times, they keep repeating the same things which fail to keep the users engaged.

2. Information-Retrieval (IR) based: Given a user input, such chatbots focus on choosing a response from a pool of unstructured conversational data. Formally, IR-based systems take a user query, q and a conversational corpus, c as input and return a response, r that is relevant to q. A number of IR algorithms [3–6] are used to rank a repository of responses in order to find the most suitable one. Although the generate response is grammatically correct, it often lacks diversity and is not within the context of the conversation. As a result, researchers have turned their attention to neural generative dialog systems.

3. Neural Generative: With the availability of large scale conversational data, many researchers looked into training and building data-driven dialog systems. Unlike IR-based systems that copy utterances from a corpus to generate response, such systems generate diverse responses by producing utterances word by word that could have never appeared together in the training dataset. Response generation can be deemed as a message-response mapping problem where the model has to learn a coherent response given previous message utterances [7].Neural generative dialog systems too have a few disadvantages. Sometimes the generated response is not semantically correct and the wide range of plausible responses can make generating the appropriate one much more difficult.

Luckily, the recent success of deep learning methods in multiple NLP tasks has spurred researchers to further investigate end-to-end dialog models [8].

A.2 History of Conversational Agents

Over the years, a lot of work has been done to transform the basic scripted QA bots to the self-learning bots we see today.

1. ELIZA: Created from 1964 to 1966 at the MIT AI Laboratory by J. Weizenbaum [1], it is the first bot that came close to competing with the Turing Test. Eliza simulated conversation by effectively recording input, rephrasing it, and matching keywords with a predefined list of responses. Because ELIZA is a rule-based system, it gave users an illusion of understanding despite having no built in framework for contextualizing events [9].

2. PARRY: Written in 1972 by psychiatrist Kenneth Colby, then at Stanford University, PARRY attempted to simulate a person with paranoid schizophrenia [2]. The program implemented a crude model of the behavior of a person with paranoid schizophrenia based on concepts, conceptualizations, and beliefs. It also embodied a conversational strategy like ELIZA but was more advanced in comparison. PARRY demonstrated how technology could assist in replicating a person with mental health issues [10].

3. ALICE: Introduced by R. Wallace in 1995, Artificial Linguistic Internet Computer Entity (ALICE) is a well-known AIML-based opensource chatbot. Inspired by ELIZA, it engages in conversations with humans by applying heuristical pattern matching rules to the user input. To make the responses more relevant and credible, supervised learning is used to track the chatbot's discussions and to suggest additional AIML content. However, because ALICE is a preset set of questions and answers, it lacks the robustness to respond to all queries [11, 12].

4. Watson: Developed by IBM in 2006, it is a retrieval-based chatbot that won the Jeopardy TV show in 2011. Watson is based on the Hadoop-based ML system and uses advanced NLP technologies including IR, Knowledge Representation and Automated Reasoning [13]. Over the past several years, the Watson Assistant chatbot has evolved and is now being deployed in different industries through fine-tuning. It even uses intent classification and entity recognition to better understand customer needs.

5. Mitsuku: Introduced by Steve Worswick in 2013, it is a rule-based chatbot written in AIML that can converse with its users with humor and empathy in a very humane way. This advanced bot won the Loebner Prize five times [14, 15]. Mitsuku's improvement includes holding long discussions, learning from chats and recalling personal information about users.

6. Siri: Released by Apple in 2011, it was originally used to assist users to perform tasks like making a call, responding to messages and managing alarms. The back-end of Siri uses Automatic Speech Recognition, NLP and other forms of weak AI to perform these tasks [16]. It also offers customization and configuration of genre, accent and language. Nowadays, it uses voice queries, gesture based control and focus-tracking to answer questions, make recommendations and perform actions. With continued use, it can adapt to user's preferences and return individualized results.

7. Alexa: Developed by Amazon in 2014, Alexa is capable of voice interaction, music playback, making to-do lists, setting alarms, playing audiobooks, providing real-time information and so on [16]. It can also act as a home automation device used to control several smart devices. Like Siri, Alexa has a natural voice and can speak with users in different languages.

8. Cortana: Developed by Microsoft in 2014, Cortana is a virtual assistant that uses the Bing search engine to perform tasks like setting reminders and answering questions for the user. With the help of Windows 10 and Windows Mobile, Cortana can perform the same duties as Siri and Alexa [16]. However, unlike its competitors, Microsoft slowly began reducing the prevalence of Cortana.

9. Google Assistant: Developed by Google in 2016, Google Assistant is a virtual assistant that is primarily available on mobile and home automation devices. Because of AI, Google Assistant can engage in two-way conversations with its user. Although keyboard input is supported, users mostly interact with it through natural voice. Like Siri and Alexa, it can answer questions, schedule events and alarms, adjust hardware settings on the user's device, show information from the user's Google account, play games and more [16].