2023

Detecting Team Conflict From Multiparty Dialogue

Ayesha Enayet
University of Central Florida

DETECTING TEAM CONFLICT FROM MULTIPARTY DIALOGUE

by

AYESHA ENAYET
M.S. Balochistan University of Information Technology, Emgineering & Management Sciences, 2016
B.S. University of Balochistan, 2011

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, FL

Summer Term
2023

Major Professor: Gita Sukthankar

ABSTRACT

The emergence of online collaboration platforms has dramatically changed the dynamics of human teamwork, creating a veritable army of virtual teams composed of workers in different physical locations. The global world requires a tremendous amount of collaborative problem solving, primarily virtual, making it an excellent domain for computer scientists and team cognition researchers who seek to understand the dynamics involved in collaborative tasks to provide a solution that can support effective collaboration. Mining and analyzing data from collaborative dialogues can yield insights into virtual teams' thought processes and help develop virtual agents to support collaboration. Good communication is indubitably the foundation of effective collaboration. Over time teams develop their own communication styles and often exhibit entrainment, a conversational phenomenon in which humans synchronize their linguistic choices.

This dissertation presents several technical innovations in the usage of machine learning towards analyzing, monitoring, and predicting collaboration success from multiparty dialogue by successfully handling the problems of resource scarcity and natural distribution shifts. First, we examine the problem of predicting team performance from embeddings learned from multiparty dialogues such that teams with similar conflict scores lie close to one another in vector space. We extract the embeddings from three types of features: 1) dialogue acts 2) sentiment polarity 3) syntactic entrainment. Although all of these features can be used to predict team performance effectively, their utility varies by the teamwork phase. We separate the dialogues of players playing a cooperative game into stages: 1) early (knowledge building), 2) middle (problem-solving), and 3) late (culmination). Unlike syntactic entrainment, both dialogue act and sentiment embeddings effectively classify team performance, even during the initial phase.

Second, we address the problem of learning generalizable models of collaboration. Machine learn-

ing models often suffer domain shifts; one advantage of encoding the semantic features is their adaptability across multiple domains. We evaluate the generalizability of different embeddings to other goal-oriented teamwork dialogues. Finally, in addition to identifying the features predictive of successful collaboration, we propose multi-feature embedding (MFeEmb) to improve the generalizability of collaborative task success prediction models under natural distribution shifts and resource scarcity. MFeEmb leverages the strengths of semantic, structural, and textual features of the dialogues by incorporating the most meaningful information from dialogue acts (DAs), sentiment polarities, and vocabulary of the dialogues.

To further enhance the performance of MFeEmb under a resource-scarce scenario, we employ synthetic data generation and few-shot learning. We use the method proposed by [7] for few-shot learning from the FsText python library. We replaced the universal embedding with our proposed multi-feature embedding to compare the performance of the two. For data augmentation, we propose using synonym replacement from collaborative dialogue vocabulary instead of synonym replacement from WordNet. The research was conducted on several multiparty dialogue datasets, including ASIST, SwDA, Hate Speech, Diplomacy, Military, SAMSum, AMI, and GitHub.

Results show that the proposed multi-feature embedding is an excellent choice for the meta-training stage of the few-shot learning, even if it learns from a small train set of size as small as 62 samples. Also, our proposed data augmentation method showed significant performance improvement. Our research has potential ramifications for the development of conversational agents that facilitate teaming as well as towards the creation of more effective social coding platforms to better support teamwork between software engineers.

To my amazing parents

and teachers.

ACKNOWLEDGMENTS

I am grateful to all the people who have supported me during my Ph.D. I am blessed to receive technical and moral support from my mentors, friends, family, and colleagues. I want to mention some of them below explicitly.

First, I would like to thank Dr. Gita Sukthankar for her valuable advice and support. Her knowledge and feedback are among the main reasons behind my achievements during my Ph.D. studies. During my Ph.D., I passed through many challenging times. She has been a great mentor who always knows what role she needs to play during the specific stage of my research. I have seen her switching between the role of tough advisor and a supporting collaborator, depending on what is required to make me more productive in my research. I could not thank her enough for the flexibility and freedom she provides me to carry on my research effectively.

I want to extend my sincere thanks to my dissertation committee members Dr. Fei Liu, Dr. Ivan Garibay, and Dr. Shawn Burke for advising me in my research. Their valuable suggestions and feedback have been extremely helpful.

I am grateful to the ASIST and SocialSim research groups. Their work gave me insight into many interesting research questions. The members of these teams are very talented and knowledgeable, and working with them was a great honor for me.

I could not thank my family enough, specifically my parents, whose support gave me the strength to work towards my goals. I am thankful to them for trusting my abilities and encouraging me to pursue Ph.D. studies.

Last but not least, special thanks to my colleagues from whom I learned many skills that helped me during my Ph.D.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

This chapter includes some content from the paper titled ”Enayet, A., & Sukthankar, G. (2021). Learning a Generalizable Model of Team Conflict from Multiparty Dialogues. International Journal of Semantic Computing, 15(04), 441-460.”.

The aim of this dissertation is to introduce new techniques to predict collaborative task success from the communication patterns between the team members. One key problem is detecting disagreement or conflict between team members. To be most useful, an agent should be able to identify the conflict at the early stages of the task in order to assist the team.

Conflict in teams can be classified as being relationship or task-oriented [113]. *Relationship conflict* arises from “interpersonal incompatibility among members, which typically includes tension, animosity, and annoyance among members within a group” [43]. Our work centers on *task conflict*, “disagreement among group members about the content of the tasks being performed, including differences in viewpoints, ideas, and opinions” [43].

Ideally, conflict prediction should be done using a very short behavior sample: “thin-slicing”. Ambady and Rosenthal demonstrate that many types of social interactions remain sufficiently stable that even a small sample is meaningful at predicting long term outcomes, the most famous application of this theory being thin-slicing marital interactions to predict divorce outcomes [5, 6]. Jung suggests that developing this capability would remove the need for developing continuous team monitoring systems [46].

Another important desideratum is to be able to generalize models across team tasks in order to handle the problem of resource scarcity. This dissertation investigates the generalizability of different dialogue features in predicting task conflict. A generalizable model is better suited for domain

adaptation scenarios in which the model is trained on multiparty dialogue data from one team task and transferred to a different task. Embeddings are mechanisms for mapping high-dimensional spaces to low-dimensions while only retaining the most effective representations, making it possible to apply machine learning on large inputs by representing them in the form of sparse vector. Unfortunately, there is a paucity of high quality data on team communications. Thus it is beneficial to learn *generalizable* embeddings that are applicable across multiple datasets. We seek to learn embeddings that are less vulnerable to domain shift in collaborative dialogues to increase the generalizability of performance prediction models.

Rather than developing specific measures for predicting future team conflict, we demonstrate that an embedding grouping teams with similar conflict levels can be learned directly from the multiparty dialogue. An advantage is that this approach avoids the necessity of collecting advance data on team members, such as personality traits or training records.

Learning a generalizable embedding involves the identification of domain invariant features. We compare the performance of three types of embeddings extracted from 1) dialogue acts, 2) sentiment polarity and 3) syntactic entrainment; these features were selected based on previous work on team communications and group problem-solving. Dialogue acts capture the interactive pattern between speakers in multiparty communication [35]. During dialogue act classification, utterances are grouped according to their communication purpose. We believe that teams who frequently engage in arguments have very different dialogue act sequences than teams who agree on the future course of action.

Sentiment polarity measures the attitude or emotion of the speaker during conversation; it can be used to detect disagreement. Entrainment is the natural tendency of the speakers to adopt a similar style during a conversation, causing them to achieve linguistic alignment. There are several types of entrainment including lexical choice [91], style [18], pronunciation [83], and many

others [74]. Reitter and Moore demonstrated that syntactic entrainment, based on alignment of lexical categories, can be used to predict success in task-oriented dialogues [91].

Good team communication exhibits all these characteristics: greater emphasis on problem solving than arguing, positive sentiment, and communication synchronization [131]. In our research, we primarily use the Teams corpus [61] which consists of player dialogue during a cooperative game. One advantage of studying a clearly defined, time-bounded team task is that the dialogues can be divided into teamwork phases: 1) early (knowledge building) 2) middle (problem solving) and 3) late (culmination). For thin-slicing, we seek to predict the team performance from the initial teamwork stages. The Teams corpus includes team conflict scores, which measure the amount of disagreement that occurred during gameplay. Our hypotheses are:

H1: an embedding leveraging dialogue acts will be useful for classifying team performance at all phases since it directly detects utterances related to conflict (eristic dialogues).

H2: sentiment analysis will consistently reveal team conflict and thus be a good predictor of performance.

H3: the entrainment embedding will be predictive when the entire dialogue is considered, but will be less useful at analyzing early phases before entrainment has been established.

H4: embeddings based on sequences of dialogues acts will generalize well at predicting task conflict across datasets.

H5: dialogue structure patterns differ between different dialogue domains.

H6: leveraging the information from semantic, structural, and lexical features in predicting collaboration success can increase the generalizability of the embeddings even if learned from a small set of samples.

Problem Statement

The goal of this research is twofold: 1) we aim to design a technique that can proactively identify the task conflict between the collaborators to provide timely assistance, and 2) we want to improve the generalizability of conflict prediction models under resource-scarce scenarios. This study makes three research contributions towards these overarching goals.

Encoding team communication with embeddings: This study compares different methods of predicting team conflict. The first approach is to generate embeddings from sequential utterance patterns. In our experiments, the multiparty dialogue is converted either to a sequence of dialogue acts or sentiments which is then used to generate the embedding. These embeddings represent meaningful information about how the communication between the team members is evolving. The second approach is to create an embedding that encodes entrainment relationships between team members. To do this, we map the whole multiparty dialogue to a feature vector representing entrainment in the teams by employing the method proposed by Rahimi et al. [89].

Conflict prediction during initial teamwork phases: During task completion, teams pass through different cognitive phases, starting from brainstorming and completing with problem solving. We compare the performance of different embeddings over teamwork phases: 1) knowledge discovery 2) problem solving and 3) culmination. We show that the sequential embeddings (dialogue act and sentiment) perform well at predicting conflict even during early teamwork phases.

Generalizability across datasets: Supervised machine learning models trained on one dataset, often do not perform well on unseen datasets; this phenomenon is called domain shift [104, 103, 122]. We test models learned on the Teams corpus on datasets gathered from software engineers (GitHub issue comments) and military teams to provide intuition on the generalizability of the three embeddings on unseen datasets. After identifying the generalizable features, we propose a method

to increase the generalizability of conflict prediction models on unseen collaborative dialogues by leveraging the strength of those features.

Research Questions

This dissertation introduces a technique to predict the success of collaborative tasks from the communication patterns between the team members. We compare different dialogue features that are predictive of team conflict. Then we demonstrate the generalizability of the features under natural distribution shifts and propose a method to improve the generalizability under resource scarcity and natural distribution shifts. We aim to answer the following research questions:

- Which features of collaborative dialogue are more predictive of task conflict?
- Which features of the collaborative dialogue are more predictive of the task conflict at different dialogue stages?
- Which features of collaborative dialogues generalize well to other datasets?
- How can we improve the generalizability of task conflict prediction models under the resource scarcity problem?
- Is there any structural similarity between and within different dialogue domains?

CHAPTER 2: RELATED WORK

This chapter includes content from the papers titled "Enayet, A., & Sukthankar, G. (2020). A transfer learning approach for dialogue act classification of github issue comments. Poster presented at the 12th International Conference on Social Informatics.", "Enayet, A., & Sukthankar, G. (2023, May). Improving the Generalizability of Collaborative Dialogue Analysis With Multi-Feature Embeddings. In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (pp. 3533-3547).", "Enayet, A., & Sukthankar, G. (2022, June). An Analysis of Dialogue Act Sequence Similarity Across Multiple Domains. In Proceedings of the Thirteenth Language Resources and Evaluation Conference (pp. 3122-3130).", and "Enayet, A., & Sukthankar, G. (2021). Learning a Generalizable Model of Team Conflict from Multiparty Dialogues. International Journal of Semantic Computing, 15(04), 441-460.".

This chapter first presents previous work on team communication analysis. Then we review prior work on the applications of dialogue act (DA) classification, sentiment analysis, and entrainment. Finally we review the most commonly used embedding models and few-shot learning.

Team Performance Analysis

Team communication, both spoken or written, is a critical element of collaborative tasks and can be studied in a variety of ways. Semantic analysis centers on the meaning of utterances, while pragmatics involves identifying speech acts [11]; both analytic approaches are important and often occur in parallel. In many studies of team communication, this analysis is arduously done through hand coding the utterances.

Parsons et al. [85] contrast two different schemes to code utterances in team dialogues as part of

their long term research goal of developing a virtual assistant for human teams. Their comparison illustrates the benefits and problems of the Walton and Krabbe typology [119], which includes categories for information-seeking, inquiry, negotiation, persuasion, deliberation, and eristic, but does not consider the context in which the utterance occurs. The McGrath theory of group behavior [68] focuses on modes of operation: inception, problem-solving, conflict resolution, and execution. When applying the McGrath theory of group behavior, utterance classification is modified by conversational context.

Sukthankar et al. also used an explicit team utterance coding scheme towards the problem of agent aiding of ad hoc, decentralized human teams to improve team performance on time-stressed group tasks [110]. Unlike teamwork studies, we do not specifically map individual utterances to team communication categories but leverage dialogue act classification models to identify features that are indicative of team conflict. Shibani et al. [106] discussed some of the practical challenges in designing an automated assessment system to provide students feedback on their teamwork competency: 1) dialogue pre-processing, 2) assessing teamwork chat text, and 3) classifying teamwork dimensions. They evaluated the performance of rule-based systems vs. supervised machine learning (SVM) at classifying coordination, mutual performance monitoring, team decision making, constructive conflict, team emotional support, and team commitment. Even with dataset imbalance, the SVM model generally outperformed the hand coded rules. Our proposed method can also be used to assist human teams by proactively warning them of deficiencies during the early phases of team tasks, without the onerous data labeling requirements.

Other analytic techniques focus on linguistic coordination between speakers in groups. For instance, Danescu et al. studied the effect of power differences on lexical category choices during goal-oriented discussion [19]. This is one form of entrainment in which the speakers preferentially select function-word classes used by other group members. In this dissertation, we use a dataset (Teams corpus), that was created to study entrainment in teams [61]. Rahimi and Litman demon-

strated a method for learning an entrainment embedding to predict team performance [89]; we use a modified version of their technique to express syntactic entrainment. However since entrainment develops over time, we compare the performance of entrainment at early vs. late task phases. Furthermore, they only focused on syntactic/lexical features of utterances, not semantic.

Sentiment analysis has been applied to the study of group dynamics; for instance, researchers have leveraged sentiment features to detect communities in social networks [101, 128]. Our work demonstrates the utility of sentiment features towards predicting team conflict and show that the sentiment-based embedding is useful during all teamwork phases. We rely exclusively on the multiparty team dialogues; however there have been many attempts to predict team performance using other types of multimodal features. TCdata, a team cooperation dataset, includes both audio and video recordings of teams performing cooperative tasks [63]. Liu et al. explicitly extracted 159 features from team speaking cues, individual speaking time statistics, and face-to-face interaction cues to predict team performance on this dataset.

Several studies [130, 81] have shown team member personality traits to be useful predictors of conflict and team performance. Yang et al. used individual personality traits to predict the performance of final year student project teams using neural networks [130]. Omar et al. developed a student performance prediction model that included both personality types and team personality diversity [81]. Even though these additional data sources can be highly predictive, they are rarely available in real-world team scenarios, unlike multi-party dialogue which is often self-archived to preserve organizational memory.

Dialogue Act Patterns

There are many areas where DAs and DA sequences have been leveraged for natural language understanding and communication analysis. Following is a brief literature review on applications of DA and DA sequences.

Coreference resolution involves the identification of entities in dialogues that refer to each other. Agrawal et al. [2] used dialogue acts as a semantic feature for coreference resolution in the human-bot scenario. By exploiting the question-answer sequence, they resolved the coreference between ”it” and ”that.” They mapped the SwDA dataset classes to one of five classes: Statement, Opinion, Question, Answer, and Other. The proposed approach showed an improvement of 24.8% in F1-score.

Aberdeen and Ferro [1] utilize DAs for detecting misunderstandings in human-computer dialogues. Their study identified some patterns, i.e. sequences of DAs, that are predictive of misunderstanding. They also identified the correlation between user satisfaction and DAs.

Goo and Chen [35] proposed an abstractive dialogue summarization method that leverages the information provided by the dialogue act of the utterance to support the summary generation process. Dialogue acts are one of the most effective ways to model inter-speaker interactions. The framework has four main components: 1) dialogue history encoder, 2) dialogue act labeler, 3) attention summary decoder, and 4) sentence gate. The dialogue act labeler uses an attention mechanism to predict the DAs of each utterance of the dialogue. The input to each decoder's hidden state is the previous state, previous state output, and the context vector.

Frummet et al. [33] consider the special case of DA classification and focused on the information need categories. They categorized user queries into 27 information need classes in the domain-specific scenario. Their work demonstrated that questions that do not contain the grammatical

structure of a question could be classified as a question based on the previous utterances. To categorize information needs, they utilize the aforementioned information need categories as a feature and the sequence IDs to predict the information need categories. The IDs represent the position of information needed for the task. In addition to annotation, they applied a random forest classifier on the data to predict the user's information need. The motivation behind the study is to make the conversational agent more aware of the need of the user.

Midgley and MacNish [69] introduced a method for discourse chunking, based on the tags of the utterances. They further showed that the discourse chunking helps in the DA classification of the utterances since some of the tags that appear in one chunk do not appear in another. Thus, the chunk information help improves the accuracy of DA tagging.

Ravi and Kim [90] performed analysis on students' online discussion to identify whether the conversation contains any unanswered questions. They applied speech classifier and rule-based thread profiling techniques to determine the need for assistance. Their SA classifier was based on N-gram features and SVM.

Lee et al. [59] designed a situational-based dialogue management system, which takes into account the intention (DA) of the user, utterance of the user, history, and discourse to take action. The proposed system uses predefined rules to analyze the current situation and take action.

Kumar et al. [54] performed a detailed analysis on the utility of dialogue acts in the development of a conversational model. They identified that both the discriminative and generative models benefit from the DA information for response generation. They also introduced a Siamese-based conversational model that leverages the strength of the conversation's hierarchical structure and DA information for the following utterance selection.

Schatzmann et al. [102] used DA information in the design of agenda-based simulator for training

dialogue manager. They formalized the dialogue as a transition from one DA and state to another at the semantic level. The agenda-based dialogue manager successfully simulated the real-world dialogues and helped in learning the effective dialogue policy. The reported accuracy achieved through the learned policy was around 90%.

Ultes et al. [118] present an open-source, multi-domain, statistical dialogues system PyDial to support research in the field of dialogue systems. The internal architecture of the system utilizes dialogue act features for language generation. The system's main components are policy, language generator, topic tracker, semantic decoder, and belief tracker. The belief tracker, policy, and simulator are domain-independent components, while the language generator and semantic decoder have domain-dependent functionality.

Zhao et al. [134] identified negotiation as a reasoning and language generation problem. They proposed a semi-automated negotiation dialogue system that automatically reasons about the conversation strategy and provides the user with linguistic choices to select their next utterance. The negotiation model is composed of the task phase and the social phase. During the social phase, both user and agent models have dialogue act sequences as part of the schema. They identified that successful negotiation depends on reasoning, planning, and appropriate language to improve the user's mutual understanding and trust. They used five speech act categories to train the speech act classifier; the main objective of the speech act classifier was to identify the agent task intention.

Bickmore and Schulman [10] defined a set of 109 dialogue acts and classified them into four categories. The classes represent the type of relationship agent and user share, i.e., 1) stranger/professional, 2) more than professional, 3) casual friends, and 4) close friends. Each DA represents the agent action that he wants to participate in or not. The proposed an accommodation theory-based approach to model the agent-user relationship.

Ahmadvand et al. [3] developed a Contextual Dialogue Act classifier (CDAC) model for the open-

domain human-machine conversational agent. The CDAC employs m previous DAC predictions and the current utterance embedding for DA classification of an utterance. The m controls the length of context. The proposed method exploits the strength of the transfer learning approach by training on human-human conversation (SwDA) and fine-tuning on human-machine conversation. They collected 200 human-machine conversations during the Amazon Alexa Prize competition in 2018 and manually labeled them with the help of human annotators.

Griol et al. [37] simulated the user and agent to collect a dialogue corpus. The language generation starts with giving equal probabilities to the possible responses. After completing the dialogue session, the probabilities of the responses adopted during the session increase, and updated values are used for the next session. Initial random selection is made based on the dialogues acts, which represent the semantics of the task.

Montenegro et al. [75] identified the need for defining a DA taxonomy, that takes into account the coaching goals, for a virtual coaching agent for the elderly. They emphasized that the coaching agents are different from other task-oriented agents and open domain conversational agents. Defining a coaching goals-oriented taxonomy could improve the performance of the dialogue manager of the coaching agent. They defined four types of labels: topic, intent, polarity, and entity labels. The context of an utterance decides the topic label of the utterance.

Milhorat et al. [72] introduced a dialogue system for Erica, an Android robot. The objective of Erica's dialogue system is to make Erica converse in a more human-like way by embodying features like backchannel, fillers, and turn-taking. The architecture of Erica is an integration of four dialogue components: question-answering, statement response, backchannel, or proactive initiator. Initially, the question-answering statement response components deal with the incoming utterance by assigning a confidence score, and then the controller selects the response associated with the highest confidence score. Both the parts compute confidence scores based on the dialogue

act tagging. The SVM-based DA tagger classifies the dialogue as either question or non-question.

Ryan et al. [95] identified the labor-intensive nature of authoring branching dialogues and the limitations it poses on the linguistic choices of the user. As an alternative to the branching approach, they proposed a procedural approach to solving the problem by introducing a policy-based system for dynamic dialogue selection. They employed dialogue moves, a variant of speech acts that incorporates more fine-grained acts representing the low-level moves of the users. The dialogue moves worked as a planning operator for achieving conversational goals.

Dialogue Act Classification

Webb et al. [126] used an n-gram model for the DA classification. They evaluated unigram, bigram. trigram, and 4-gram models; based on the criteria of predictivity, they selected the n-grams as cue phrases. The predictivity of a cue phrase represents how predictive the specific n-gram is in detecting the DA. The highly predictive n-grams were then used for the classification.

Grau et al. [36] applied a naive Bayes classifier along with 2-grams and 3-grams on two different corpora for dialogue act classification.

Ezen-Can et al. [30] proposed an unsupervised multimodel feature-based technique for the DA classification of student dialogues. They employed lexical, dialogue-context, task, and posture + gesture-based features. The dialogue-context feature set includes the DA of the previous tutor utterance.

Ezen-Can et al. [30] applied data mining based unsupervised DA classification approach on education domain, called Markov Random Field (MRF). They performed experiments on tutorial dialogue corpus collected from an introductory Java programming project. MRF works similar to

query likelihood clustering but also considers word ordering.

Chen and Di Eugenio [15] utilizes the multimodal corpus, including haptic action and pointing gesture, to enhance the DA classification performance. In addition to multimodal features, they also employed a dialogue game feature. The dialogue game feature incorporates a hierarchical structure of dialogue and improves the performance significantly.

Li et al. [60] applied a dual-attention hierarchical RNN for the classification of dialogues. They identified the significant conceptual relation between DA and topic and used topic identification as an auxiliary task. They explained that while DA represents the social act, the topic defines the subject under consideration, directly related to the type of DA's that could occur in the conversation.

Serafin and Di Eugenio [105] performed dialogue act classification using the feature LDA technique. They augmented an LDA vector with features such as the Part of Speech (POS) tag to assign a DA tag to the utterance. This reduced the error rates up to 60% to 78%.

Tran et al. [115] employed a generative neural network model that defines a joint probability distribution over a sequence of DAs and utterances. The generator generates the current DA based on the previous DA and current utterance and current utterance based on the current DA and previous utterance.

Milajevs and Purver [71] presented an analysis over three different DA tagging modeling approaches, which include: 1) bag of word model, 2) word distribution-based model, and 3) utterance order based modeling. They identified the limitations of a bag of word model and used it as a baseline model. Their proposed models exploited the intra-utterance word order and word co-occurrence information and performed significantly better than the bag of words approach.

Tran et al. [116] proposed a neural network-based approach, similar to hidden Markov model, that uses the neural network's probability distribution over current label as an input for the next step.

Sentiment Analysis

Sentiment analysis plays an important role in revealing the emotional state and emotional changes of the speaker involved in conversation [133]. This section gives a brief literature review of the applications of sentiment classification.

Sun et al. [112] proposed a method for anomaly detection in conversation from the emotional transition patterns of the speakers. The study provides a framework that combines the convolutional neural network long short-term memory (CNN-LSTM) with a Markov chain Monte Carlo (MCMC) to track the dynamic transition in the speaker's emotional state. A similarity function compares the speaker's historical emotional transition tensor, i.e., normal transition tensor, with the current emotional transitional tensor to identify the anomaly.

Fraser et al. [32] used sentiment analysis to enhance player engagement in role-playing video games. The dialogue manager of the system uses the IBM Watson's Tone Analyser and Persona's AIML patterns to identify the player's emotional state and controls when and what information is to provide to the player.

Wang et al. [120] performed sentiment classification on customer service dialogues. They proposed a multi-task learning-based sentiment classification approach. They identified that using topic modeling as an auxiliary task could improve the sentiment classification performance of Customer Service dialogues. In addition, they employed an attention mechanism to learn context-aware representations of the utterances.

Sentiment analysis has also been applied to mental health monitoring systems [94], restaurant recommendation systems [111], movie recommendation systems [125], and E-commerce recommendation systems [107]. Social media analysis is one of the most significant applications of sentiment analysis. Researchers have performed sentiment analysis on many social media platforms

including Twitter [66, 121, 80, 98], Facebook [99, 93, 87, 39], and Instagram [109, 79].

Entrainment

Rahimi and Litman [89] proposed a graphical method to encode the entrainment between the team members in multiparty dialogue. They identified that entrainment could predict good performance and teamwork.

Flemotomos et al. [31] studied the relationship between entrainment and dominance in multiparty communication dynamics. The study identified that the dominant speaker is less likely to adopt the style of other members, while the least dominant speakers change their linguistic choices and align them with dominant speakers. They applied a multimodal approach to measuring the entrainment.

Beňuš et al. [9] analyzed the correlation between prosodic entrainment and trust in human-computer interaction. The study identified that females show more trust toward disentraining avatars.

Lubold [64] identified the importance of acoustic-prosodic entrainment in building rapport in spoken dialogue systems. The study was performed on robotic learning companions and contributed to developing an agent that can entrain to improve the student's learning outcome and identified the correlation between entrainment and learning outcome.

Embeddings

Embeddings are a mechanism for mapping a high-dimensional space to a low-dimensional one while only retaining the most effective structural representations. They can be used as part of the transfer learning process to mitigate the low availability of labeled language resources on various NLP tasks. Some of the most popular embedding methods are Global Vectors for Word Rep-

resentation (GloVe) [86], Universal Sentence Encoding (USE) [14], and Bidirectional Encoding (BERT) [22] for sentence representation.

Global Vectors for Word Representation (GloVe)

Pennington et al. [86] proposed the GloVe model in 2014. It creates a word-level embedding that leverages both the local context window and global matrix factorization methods. GloVe employs a log-bilinear prediction-based technique that utilizes word-word co-occurrence statistics to identify a meaningful structure and generate word-level embeddings.

Universal Sentence Encoders (USE)

In 2018, Google Research released a Universal Sentence Encoder (USE) model for sentence-level transfer learning that achieves consistent performance across multiple NLP tasks [14]. There are two different variants of the model: 1) a transformer architecture, which gives high accuracy at the cost of high resource consumption and 2) a deep averaging network that requires few resources and makes small compromises for efficiency. The former uses attention-based, context-aware encoding sub-graphs for the transfer architecture. The model outputs a 512-dimensional vector. The deep averaging network works by averaging words and bigram embeddings to use as an input to a deep neural network. The models are trained on web news, Wikipedia, web question-answer pages, discussion forums, and the Stanford Natural Language Inference (SNLI) corpus.

Bidirectional Encoder Representations from Transformers (BERT)

Also created at Google, BERT is the first model that was trained on both left and right contexts [22]. To achieve pre-trained deep bidirectional representation, it uses the masked model, which follows

the cloze deletion task. This model is trained on Books Corpus and English Wikipedia corpus. The code for BERT is available at `https://github.com/google-research/bert`. There are two available flavors of BERT: 1)$BERT_{BASE}$, and 2) $BERT_{LARGE}$. $BERT_{BASE}$ has 12 transformer blocks, 768 hidden layers, 12 self-attention heads, and 110 million parameters. On the other hand, $BERT_{LARGE}$ uses a fairly large network, with 24 transformer blocks, 1024 hidden layers, 16 self-attention heads, and 340 million parameters.

Probabilistic Representation with Recurrent Neural Networks

Duran et al. proposed a probabilistic technique to represent utterances while using the LSTM sentence model for dialogue act classification [24]. The probabilistic distribution of each word in the corpus over DA categories provides the representation of the utterances. The model does not incorporate contextual features at the discourse level. The set of keywords consisting of all the words that occur above a threshold frequency is used to define a $n \times m$ matrix X, where m is the number of categories, and n is the number of keywords. Each entry x_{ij} of the matrix represents the probability of the tag j given the word i. Training code is available at `https://github.com/NathanDuran/Probabilistic-RNN-DA-Classifier`.

Few-Shot Generalizability

Triantafillou et al. [117] introduced a method that improves few-shot generalizability by making use of multiple datasets in order to learn a universal template. Dvornik et al. [25] proposed Selecting from Universal Representations (SUR), which involves learning a multi-domain representation by training multiple feature extractors. A multi-domain feature bank is used to select the most relevant feature during the learning phase.

Sauer et al. [100] introduced a method to distill knowledge from the few-shot model in which both the student and teacher networks are prototypical networks, i.e., the samples are classified by measuring the distance between the prototype representation of the class and the samples.

Few-shot generalizability is also an important area of research in the field of Computer vision. Computer vision researchers have identified many techniques to improve the generalizability at the meta-learning phase. Kozerawski and Turk [51] introduced Meta Binary Cross-Entropy (Meta-BCE) and One-Class Meta-Learning (OCML), a few-shot one-class classification method to detect out-of-distribution class in one-class and multiclass open-set problems scenarios. Meta-BCE works by learning a rich one-class representation using binary cross-entropy loss in a meta-learning setting. In this setting probability of any sample belonging to or not belonging to any specific class only depends on the positive samples of the class without considering other classes. OCML works by dynamically creating one-class neural network classifier for a new category. The transfer module of OCML helps transform the feature vector of category c to the weight vector of category c for the one-class classifier.

Dong et al. [23] introduced an adversarial-aware mechanism for few-shot image classification. A robust embedding model is proposed to learn a representation that can effectively differentiate between legitimate and adversarial examples. Jamal and Qi [42] introduce the Task-Agnostic Meta-Learning (TAML) algorithm to improve the generalizability of the few-shot classification. TAML learns an unbiased initial model by maximizing the entropy of the out labels. This prevents the model from overperforming on specific tasks. [49] proposed a method to improve the few-shot generalizability of the object detection method by extracting generalizable meta-features. The feature learning module learns generalizable features from the base class, and then reweighting module identifies the meta-features predictive of the novel class with the help of a few available samples to generate a global vector.

Das et al. [20] applied a contrastive learning approach while finetuning to achieve generalizability. They use the unlabeled examples from the based class as distractors, and the generalizability is achieved by contrasting the distractors and task-specific samples.

CHAPTER 3: DATASETS

This chapter includes content from the papers titled "Enayet, A., & Sukthankar, G. (2020). A transfer learning approach for dialogue act classification of github issue comments. Poster presented at the 12th International Conference on Social Informatics.", "Enayet, A., & Sukthankar, G. (2023, May). Improving the Generalizability of Collaborative Dialogue Analysis With Multi-Feature Embeddings. In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (pp. 3533-3547).", "Enayet, A., & Sukthankar, G. (2022, June). An Analysis of Dialogue Act Sequence Similarity Across Multiple Domains. In Proceedings of the Thirteenth Language Resources and Evaluation Conference (pp. 3122-3130).", and "Enayet, A., & Sukthankar, G. (2021). Learning a Generalizable Model of Team Conflict from Multiparty Dialogues. International Journal of Semantic Computing, 15(04), 441-460.".

This dissertation research was conducted using several different team communication datasets. Following are the details of the datasets that are part of this study.

Multiplayer Board Games (Teams Corpus)

Teams contains 124 team dialogues from 62 different teams, playing two different collaborative board games. The length of the dialogues varies from 291 to 2124 utterances. In addition to collecting dialogue data, the researchers administered surveys of team level social outcomes. Team social outcome scores include task conflict, relation conflict, and process conflict scores. All these scores are highly correlated, and we are using process conflict z-scores to represent team conflict. Table 3.1 shows the example from Teams dataset.

Table 3.1: Teams Dataset Sample

Speaker	Utterance	DA	Description
A	Ok I'm going to	sd	Statement-non-Opinion
A	shore up these two.	sd	Statement-non-Opinion
B	Good move.	ba	Appreciation
A	Then we got one and then I guess I can also	sd	Statement-non-Opinion
A	Can I use my powers twice in one play	sd	Statement-non-Opinion
C	Mm	b	Acknowledge (Backchannel)
B	yes	ny	Yes answer

Software Engineering Teams (GitHub Issue Comments)

The GitHub social coding platform is specialized to support virtual teams of software developers whose primary communication goal is to discuss new features and monitor software bugs. Our assumption is that each software repository is maintained by a team and that the events associated with the repository form a partial history of the team activities and social interactions. Within GitHub's issue handling infrastructure, users can report a bug or provide a feature request by opening an issue.

We created a dataset from software engineering teams resolving issues on GitHub which we are in the process of making publicly available at: `https://drive.google.com/file/d/17W3zeyN3EUJAMYTJVbDcPXmg6DQcqXT6/view`. Table 3.2 shows the statistics of our corpus. The length of the dialogues in our GitHub corpus varies from 2 to 207 utterances. Utterances from the GitHub dialogues, unlike the Teams corpus, are combination of English language words, special symbols, and code written in different programming languages. The average length of the dialogues is 19. The number of speakers varies from 2 to 10. While collecting the dialogues, to preserve the complex nature of the GitHub dialogue we didn't place any limitation on the total number of speakers and the length of the dialogue. Code blocks were removed if they appeared separately in the dialogue but not if they appeared within the utterance.

Table 3.2: Statistics of GitHub Issue Comments Dataset

#Dialogues	# Utterances	#Tokens	#DA tags	#Positive Samples	#Negative Samples
50	981	13418	42	29	21

Table 3.3 shows the example from GitHub corpus. Since we lack post-task process conflict survey scores from the team members, we manually labeled the dialogues as being high conflict or low conflict using the following criteria:

1. The issue did not resolve successfully.
2. The question(s) of the team member(s) remained unanswered.
3. One or more team members did not understand the issue.
4. Lack of understanding or disagreement between the team members.
5. At least one team member did not agree with the suggested solution.

This criterion is based on Kalia et al's [48] work on affective processes in teams. An affective process represents the motivational and affective relationships between the members of the team. They evaluated dyadic communication between team members including 1) responses to questions, 2) responses to directives, 3) responses to requests, 4) responses to commissives and 5) responses to informatives. The team member's response (taking the required action) to the other team member's directives and requests is an example of positive evidence indicating low conflict. The absence of the response counts as negative evidence. Response to the informatives, questions, and commissives is an example of neutral evidence.

Military Dataset

We also used Kalia et al.'s military team communication dataset [48] which contains 22 chats from 20 chat rooms. The chats are communication from simulation activity (SIMEX). The average

Table 3.3: GitHub Dataset Sample

Speaker	Utterance	DA	Description
m1	I'm following up on this SO question as no one else has. The comments recommend posting a feature request here.	sd	Statement-non-Opinion
m1	I have an R package on github. This R package has C++ dependencies which I include in a src.	sd	Statement-non-Opinion
m1	The correct way I would normally do this (outside of R) is create submodules within the github repo which could link to the correct commits as dependencies.	sd	Statement-non-Opinion
m1	So the checking for empty or unneeded directories causes the errors because the submodules are interpreted as empty subdirectories. Therefore it cannot find the necessary dependencies and I'll run into a fatal error upon build	sd	Statement-non-Opinion
m1	Yes one way to solve this is to physically put the dependencies within the R package. That does defeat the purpose of submodules though which are very useful.	aa	Agree/Accept
m1	It appears using the following argument works:	sd	Statement-non-Opinion
m1	The problem with this is this isn't default behavior. I'm nervous about getting dozens of github issues from users who ran$devtools :: install_git("reponamepackagename")$ and didn't read the fine print in the README	sd	Statement-non-Opinion
m1	Is there a better way?	qy	Yes-No-Question
m1	What is the standard method of releasing R packages as a github repo using submodules?	qw	Wh-Question
m2	FWIW there is a on-going PR for installing github repo with submodules in#103. When it is done it may answer your use case.	sv	Statement-opinion
m3	I would recommend using subtrees instead of submodules which will just work for users without any additional tooling.	sd	Statement-non-Opinion
m3	As of 0927172 remotes now automatically detects submodules and installs them as needed.	sd	Statement-non-Opinion

number of speakers in their corpus is 15, which is larger than the other two datasets. The length of the dialogue varies from 55 to 1027 utterances. Table 3.4 shows example utterances from the military dataset and their dialogue act classification. This dataset also contains post-event survey reflecting qualitative measures of team performance. Kalia et al. [48] used the meaning of the messages from broadcast communication to evaluate how the team process measures change with time; we use the post-event survey results to annotate the whole teamwork chat as being high or low conflict.

Table 3.4: Military Dataset Sample

Speaker	Utterance	DA	Description
m1	it says 34 cdr is talking in bde room.	sd	Statement-non-Opinion
m1	bandit 6 came in pretty quiet in bde room	sd	Statement-non-Opinion
m2	roger	b	Acknowledge (Backchannel)
m3	are we atlking this T72 spotted by B Co 2-44 IVO 12SWG 61768 89877?	qy	Yes-No-Question
m2	not tracking this one.	sd	Statement-non-Opinion
m3	2-44 taking small arms fire in and around Airfield West, and from OBJ 5.	sd	Statement-non-Opinion
m3	WPNs CO 2-44 engaging tech vechicle IVO 12SWG 61794 90137 and 7 dismounts	sd	Statement-non-Opinion
m1	CTRP 100%.	sd	Statement-non-Opinion
m2	roger c trp .	sd	Statement-non-Opinion

ASIST Dataset

The Artificial Social Intelligence for Successful Teams (ASIST) dataset was developed to analyze how well an artificial intelligence system can predict and infer the action of an individual in a three member team. The dataset contains communication between the team members completing an Urban Search and Rescue task in the Minecraft environment. In addition to collecting dialogue data, the researchers administered surveys of team level social outcomes. The dataset contains team process scores. We manually use team process z scores to classify teams into high conflict and low conflict teams. We apply analysis on 113 dialogues; Table 3.5 shows an example from the ASIST dataset and their DA classification.

Table 3.5: ASIST Dataset Sample

Speaker	Utterance	DA	Description
E000302	this is green	sd	Statement-non-opinion
E000302	this is green no questions thank you	sd	Statement-non-opinion
E000303	this is blue no questions	sd	Statement-non-opinion
E000301	and it's ready	sd	Statement-non-opinion
E000303	blue is ready	sd	Statement-non-opinion
E000302	green is ready	sd	Statement-non-opinion
E000303	how to do a question	qw	Wh-Question
E000303	can you overwrite your team's markers or no	qy	Yes-No-Question
E000303	a teammate's markers	sd	Statement-non-opinion
E000301	drednaw	b	Acknowledge (Backchannel)

SwDA

SwDA is one of the most popular public datasets for DA classification. It consists of 1155 human-to-human telephone speech conversations [1]. The dataset is tagged using 42 tags from the SwDA-DAMSL tagset, which is a subset of Dialogue Act Markup in Several Layers (DAMSL) categories [4]. A more detailed description of SwDA-DAMSL is provided by [47] [2]. Table 3.6 shows the example from the SwDA dataset.

SAMsum

SAMsum is a chat dialogue dataset which consists of Messenger, Whatsapp, and WeChat conversation, written and created by linguists. The dataset contains 16,369 dialogues which include 14,732 train, 819 test, and 818 validation dialogues [34]. Table 3.7 shows the example from SAMSum

[1] https://github.com/cgpotts/swda

[2] https://web.stanford.edu/~jurafsky/ws97/manual.august1.html

Table 3.6: SwDA Dataset Sample

Speaker	Utterance	DA	Description
A	I don't, I don't have any kids.	sd	Statement-non-Opinion
A	I, uh, my sister has a, she just had a baby,	sd	Statement-non-Opinion
A	he's about five months old	sd	Statement-non-Opinion
A	and she was worrying about going back to work and what she was going to do with him and –	sd	Statement-non-Opinion
A	Uh-huh.	b	Acknowledge
A	do you have kids?	qy	Yes-No-Question
B	I have three.	na	Affirmative non-yes Answer
A	Oh, really?	bh	Backchannel in question form

Table 3.7: SAMSum Dataset Sample

Speaker	Utterance	DA	Description
Hannah	Hey do you have Betty's number?	qy	Yes-No-Question
Amanda	Lemme check .	sd	Statement-non-Opinion
Amanda	Sorry can't find it.	sd	Statement-non-Opinion
Amanda	Ask Larry .	sd	Statement-non-Opinion
Amanda	He called her last time we were at the park together .	sd	Statement-non-Opinion
Hannah	I don't know him well .	sd	Statement-non-Opinion
Amanda	Don't be shy he's very nice .	sv	Statement-opinion
Hannah	If you say so. .	b	Acknowledge (Backchannel)
Hannah	I'd rather you texted him .	fc	Conventional-closing
Amanda	Just text him.	ad	Action-directive
Hannah	Urgh. Alright .	b	Acknowledge (Backchannel)
Hannah	Bye .	fc	Conventional-closing
Amanda	Bye bye	fc	Conventional-closing

Dataset.

AMI (DialSum)

DialSum, a subset of the AMI meeting corpus, contains 24,193 total dialogues, divided into 7,024 train, 400 test, and 400 validation instances. It is a subset of the AMI meeting corpus with the topic descriptions as abstractive summaries. The AMI meeting corpus contains transcriptions of 100 hours of meeting recordings [3]. Table 3.8 shows the example from AMI (DialSum) dataset.

[3] https://github.com/MiuLab/DialSum

Table 3.8: DialSum Dataset Sample

Speaker	Utterance	DA	Description
	well i suppose that's our that's that's our design that we've got . so	sd	Statement-non-Opinion
	yeah .	b	Acknowledge (Backchannel)
	yeah yeah .	b	Acknowledge (Backchannel)
	well that's that's uh	ba	Appreciation
	okay so project evaluation .	sd	Statement-non-Opinion
	we have under twelve euros fifty .	sd	Statement-non-Opinion
	project process how do we think that went ?	sd	Statement-non-Opinion
	are we happy ?	qy	Yes-No-Question

Diplomacy Betrayal

Diplomacy dataset consists of communication between online users playing the Diplomacy strategic board game. The dataset contains games with different outcomes: half of which ended in betrayal and half ended in friendship [4]. Table 3.9 shows the example from Diplomacy Betrayal dataset.

Hate Speech

Hate Speech dataset consists of utterances extracted from the posts of white supremacist forum. The sentences of the posts are annotated to reflect the presence or absence of Hate Speech [5]. Table 3.10 shows the example from Hate Speech dataset.

[4] https://sites.google.com/view/qanta/projects/diplomacy

[5] https://github.com/Vicomtech/hate-speech-dataset

Table 3.9: Diplomacy Betrayal Dataset Sample

Speaker	Utterance	DA	Description
turkey	'Hello Italy whats up what are your thoughts on austria and France?	qw	Wh-Question
italy	Hi Turkey! Im sorry that Ive been so slow to get in touch. Kind of a rough day for me to begin a game as I e been pretty swamped. Things are clearing up now, and I appreciate you reaching out to me. ¡EOS¿So far I have notes from Austria and Russia being pretty cagey and non-committal. Perhaps that is just the life of Italy? Nobody really has me in their plans?¡EOS¿I dont really know what Im going to do yet, so if you have ideas, or you have a use for me, please let me know. Id basically be delighted to work with anyone who really wants to work with me. (No sign yet that this includes anyone at all)	sd	Statement-non-Opinion
italy	Hey Turkey 2014 any interest in working together? Im trying to think of ideas, but Id just like to know first if you have interest, and then we can work out a plan if you do.	sd	Statement-non-Opinion
italy	Any thoughts, Turkey?	qo	Open-Question
turkey	Sure we can work together austria would be the most likely candidate for us to maul	sd	Statement-non-Opinion

Table 3.10: Hate Speech Dataset Sample

Speaker	Utterance	DA	Description
	Thank you for posting your story .	fc	Conventional-closing
	I think you should write a book as well .	sv	Statement-opinion
	I 've always considered teaching as one of the professions I would like to get into , but not in a neighbourhood like that ... never. kids like that disgust me .	sd	Statement-non-Opinion
	And the sad thing is the white students at those schools will act like that too .	sv	Statement-opinion
	I guess I 'll just stick to homeschooling my kids , when and if I have them ...	sd	Statement-non-Opinion

CHAPTER 4: TRANSFER LEARNING BASED DIALOGUE ACT CLASSIFIER

This chapter includes contents and figures from the paper titled ”Enayet, A., & Sukthankar, G. (2020). A transfer learning approach for dialogue act classification of github issue comments. Poster presented at the 12th International Conference on Social Informatics.”.

Analyzing the dialogue between team members can yield important insights into the performance of virtual teams. As a part of this dissertation, we present a transfer learning approach for performing dialogue act (DA) classification on the dialogues of the virtual teams. DA classification is the process of identifying the speaker's intent, and it plays an essential role in the semantic analysis of dialogues. Since no large labeled corpus of virtual teams communication is available, we collect utterances from GitHub issue comments and employ transfer learning for the DA classification of GitHub issue comments. Transfer learning enables us to leverage standard dialogue act datasets to label collaborative dialogues. We compare the performance of word and sentence level encoding models, including Global Vectors for Word Representations (GloVe), Universal Sentence Encoder (USE), and Bidirectional Encoder Representations from Transformers (BERT). This helps us develop and identify a DA classification model under resource scarcity scenario. Being able to map the issue comments to dialogue acts is a helpful stepping stone towards understanding cognitive team processes. We use the best performing model for our collaborative dialogue analysis.

GitHub

GitHub is a social coding platform where people collaborate virtually to propose solutions related to various software related issues. Software engineering requires a tremendous amount of collabo-

rative problem solving, making it an excellent domain for team cognition researchers who seek to understand the manifestation of cognition applied to team tasks. Mining data from social coding platforms such as GitHub can yield insights into the thought processes of virtual teams. We treat GitHub as our test case to identify a high-performing transfer learning based DA classification model. Previous work on issue comments [38, 77, 82] has focused on emotional aspects of team communication, such as sentiment and politeness. Our aim is to map issue comments to states in team cognition such as information gathering, knowledge building and problem solving. To do this we employ dialogue act (DA) classification, in order to identify the intent of the speaker.

Dialogue act classification has a broad range of natural language processing applications, including machine translation, dialogue systems and speech recognition. Semantic-based classification of human utterances is a challenging task, and the lack of a large annotated corpus that represents class variations makes the job even harder. Compared to the examples of human utterances available in standard datasets like the Switchboard (SwBD) corpus and the CSI Meeting Recorder Dialogue Act (MRDA) corpus, GitHub utterances are more complex.

We perform the DA classification of GitHub issue comments by harnessing the strength of transfer learning, using word and sentence level embedding models fine-tuned on our dataset. For word-level transfer learning, we have used GLoVe vectors [86], and Universal Sentence Encoders [14] and BERT [22] models were used for sentence-level transfer. we present a comparison of the performance of various architectures on GitHub dialogues in a limited resource scenario. One of our contributions is our publicly available dataset of annotated issue comments. In the field of computational collective intelligence, where people collaborate and work in teams to achieve goals, dialogue act classification can play a vital role in understanding human teamwork.

Background (GitHub)

Unlike general purpose communication platforms such as Twitter and Facebook, GitHub is specialized to support virtual teams of software developers whose primary communication goal is to discuss new features and monitor software bugs. It facilitates distributed, asynchronous collaborations in open source software (OSS) development. Code development, issue reporting, and social interactions are tracked by the 20+ event types. Our assumption is that each software repository is maintained by a team and that the events associated with the repository form a partial history of the team activities and social interactions.

GitHub has an open API to collect metadata about users, repositories, and the activities of users on repositories. Developers make changes to the code repository by pushing their content, while GitHub tracks the version control process. Any GitHub user can contribute to a repository by sending a pull request. Repository maintainers review pull requests, discuss possible modifications in the comments, and decide whether to accept or reject the requests. GitHub also supports passive social media style interactions such as following repositories or developers. Within GitHub's issue handling infrastructure, users can report a bug or provide a feature request by opening an issue. Issue closure rates thus reflect the speed with which teams resolve problems and can be used as a measure of team performance.

Related Work (GitHub)

Issue resolution has been viewed by many researchers as a rich source of information about the emotional health of the team and how it affects the software development process [41]. Kikas et al. demonstrated a model for predicting issue lifetime that included a single feature aggregating textual comment information [50]. Several studies have employed sentiment analysis [38, 77, 82]

and topic modeling [123] to study GitHub issue comments. Ortu et al. conducted a large study on communication patterns in which they measured politeness and emotional affect in issue comments; their aim was to understand how contribution levels modulate communication patterns [82]. Murgia et al. demonstrated a machine learning classifier for identifing love, joy or sadness in issue comments [77]. An empirical study of issue comments conducted by Guzman et al. [38] showed that the sentiment expressed in issue comments varies based on day of week, geographic dispersion of the team, and the programming language. Yang et al. addressed the more practical question of the relationship of issue comment sentiment and bug fixing speed [129].

We aim is to study the team cognition aspects of collaborative problem solving using dialogue act classification. Unlike topic modeling or sentiment analysis, dialogue act classification has not been extensively applied to GitHub data. However, Saha et al. [96] proposed a deep learning approach for the dialogue act classification of Twitter data. A convolutional neural network was used to create the classifier, along with hand-crafted rules. Seven classes were included: statement, expression, suggestion, request, question, threat, and other. In contrast, our work is done using a transfer learning approach and a significantly larger set of classes.

Prior to deep learning, statistical approaches such as hidden Markov models, have been used for dialogue act classification. The HMM represents discourse structure, with dialogue acts as states. Stolcke et al. demonstrated such a model that combined prosodic, lexical, and collocational cues [108]. Chen et al. proposed the CRF-Attentive Structured Network (CRF-ASN) framework to exploit the CRF-attentive structure dependencies along with end-to-end training [17].

We presents a transfer learning approach for dialogue act classification that is used to compensate for our small dataset. To do this, we learn an embedding from a larger dataset.

Table 4.1: Dataset Statistics

Dataset	Categories	#Utterances	#Tokens
SwDA	42	200,052	19K
GitHub	42	859	10,131

Datasets

For our study, we collected a dataset of issue comments from GitHub and hand annotated them using a standard dialogue act tagset, DAMSL (Discourse Annotation and Markup System of Labeling), to facilitate the transfer process. The tagset is available at `https://web.stanford.edu/~jurafsky/ws97/manual.august1.html`. Our test set consists of 859 instances from more than 50 GitHub issues.

The models were trained using the Switchboard Dialogue Act Corpus (SwDA) dataset. SwDA is one of the most popular public datasets for DA classification. It consists of 1155 human-to-human telephone speech conversations. The dataset is tagged using 42 tags from the DAMSL tagset. Table 4.1 shows the statistics of both test and train datasets. Table 3.6 shows examples from the SWDA training dataset. Table 3.3 shows examples from our GitHub issue comment dataset. From these examples, it is clear that this is a challenging transfer learning problem.

Method

Treating our dialogue act classification as a transfer learning problem enables us to leverage embeddings learned on a dataset that is over 200 times larger than our test dataset. We created and evaluated several different dialogue act classification pipelines using five different architectures and four different word and sentence-level embedding models. Figure 4.1 illustrates the differences in

architecture between our five models.

Probabilistic Representation with Recurrent Neural Networks

Duran et al. proposed a probabilistic technique to represent utterances while using the LSTM sentence model for dialogue act classification [24]. The probabilistic distribution of each word in the corpus over DA categories provides the representation of the utterances. The model does not incorporate contextual features at the discourse level. The set of keywords consisting of all the words that occur above a threshold frequency is used to define a $n \times m$ matrix X, where m is the number of categories, and n is the number of keywords. Each entry x_{ij} of the matrix represents the probability of the tag j given the word i. Training was accomplished using code downloaded from `https://github.com/NathanDuran/Probabilistic-RNN-DA-Classifier`.

GloVe + LSTM

We use glove.6b.100d.txt downloaded from `https://nlp.stanford.edu/projects/glove/` to train our model on the SwDA dataset. The model consists of input, embedding, LSTM, and one dense layer with 42 output labels and a softmax activation function.

Universal Sentence Encoder (USE)

Results were obtained using the USE model from TensorFlow Hub, after fine-tuning on the SwDA dataset. The code to load the USE model is available at `https://tfhub.dev/google/universal-sentence-encoder/1`. We chose the USE Transformer-based Architecture model with three dense layers and a softmax activation function.

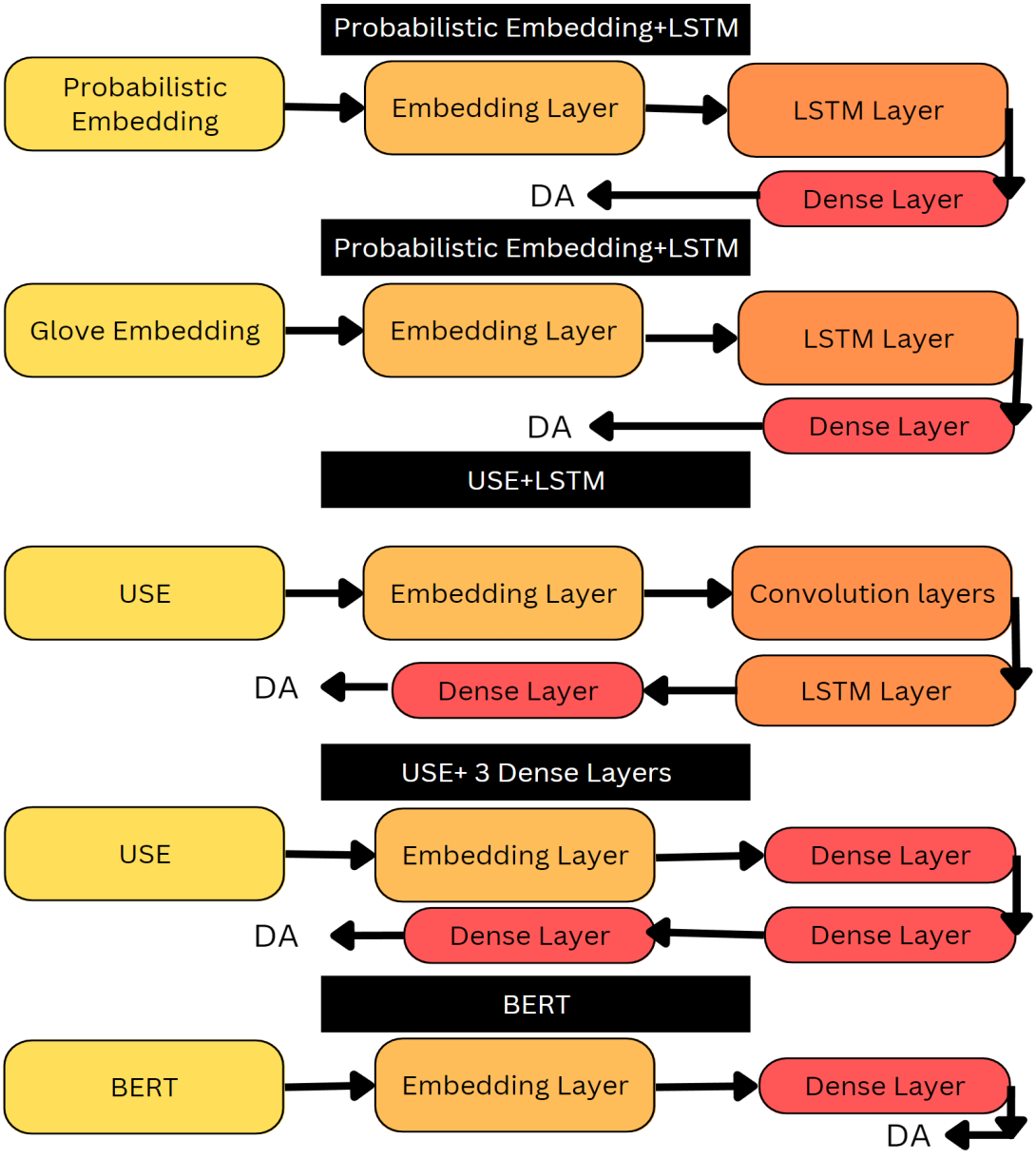


Figure 4.1: Architecture Diagrams for (i) Probabilistic Representation with RNN (ii) GloVe+LSTM (iii) Universal Sentence Encoder (USE) (iv) USE+LSTM (v) Bidirectional Encoder Representations from Transformers (BERT) Architectures

USE+LSTM

We also combined the Universal Sentence Encoder with an LSTM. This model consists of Input, Embedding, Convolution, LSTM, and one Dense output layer.

Bidirectional Encoder Representations from Transformers (BERT)

As proposed by [22], we append a single dense layer to BERT. Our implementation was created using the Python TensorFlow-Bert module.

Evaluation

Table 4.2 shows the performance of all five architectures. Universal Sentence Encoder had the best performance on the GitHub issue comments, with a test accuracy of 50.71% which is 6% more than the accuracy achieved using the probabilistic representation of sentence. The other three models showed significantly lower performance than USE, lagging by almost 10%. The probabilistic representation of sentence approach exhibited the highest validation accuracy of 76.9% which is significantly higher than USE which had a validation accuracy of 69.5%. The well-known BERT model had a validation accuracy of 71.5%, but had a low test accuracy.

It is instructive to examine the performance differences between the best (USE) and second best (probabilistic representation). Figure 4.2 shows the confusion matrix of the classification results obtained using the USE model, and Figure 4.3 shows the confusion matrix of the probabilistic representation method. In both cases the most confused tag pair is sd (statement-non-opinion) & sv (statement-opinion). USE correctly classified 91.71% of the sd occurrences, while the probabilistic representation method only classified 76% correctly. On the other hand USE classified 38.98% sv

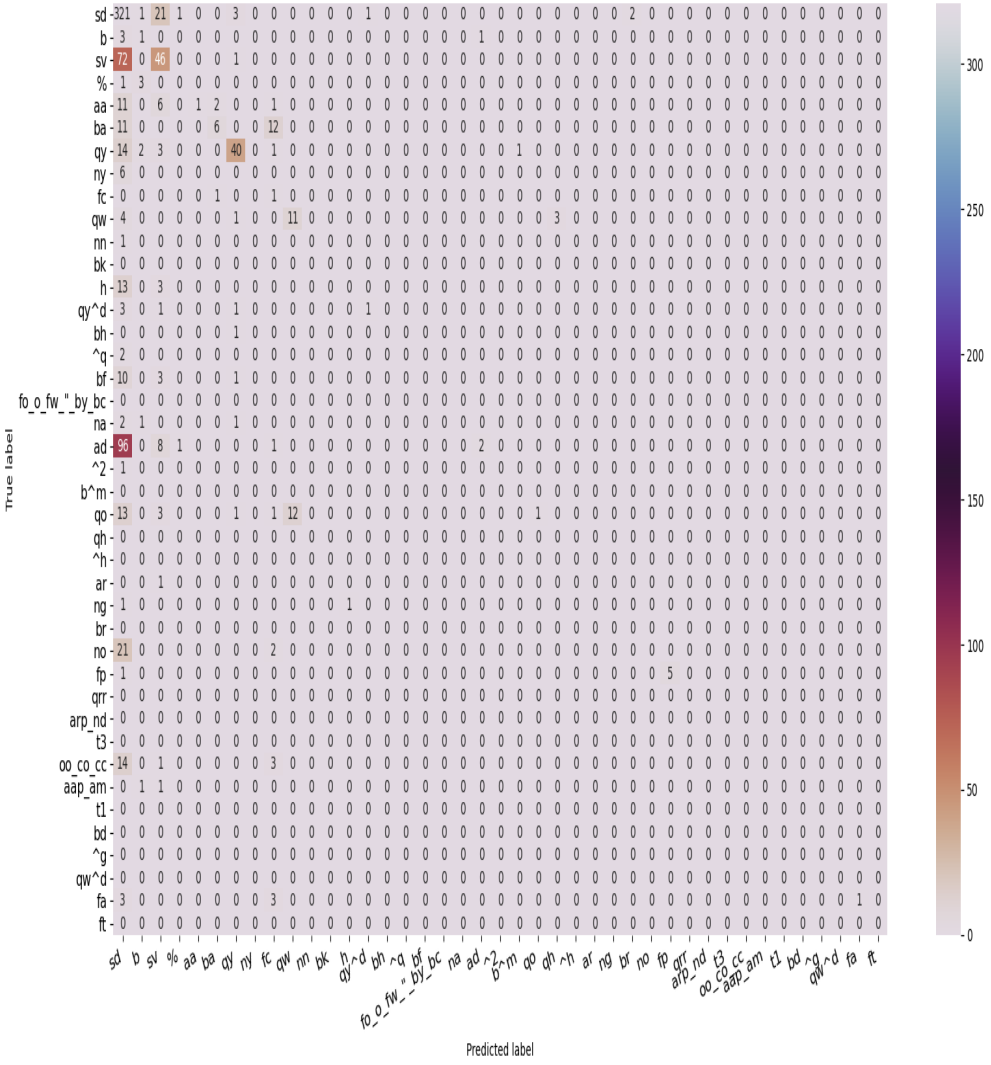


Figure 4.2: Confusion Matrix: Universal Sentence Encoder (all classes)

utterances correctly while the probabilistic representation method classified 50% of them correctly. Table 4.3 shows the Precision, Recall, & F_1 Score of our best performing architecture. In Table 4.3, support represents the number of occurrences of each tag in the GitHub dataset. USE failed to classify the third most frequent tag i.e. ad (action directive) in the test dataset. The average

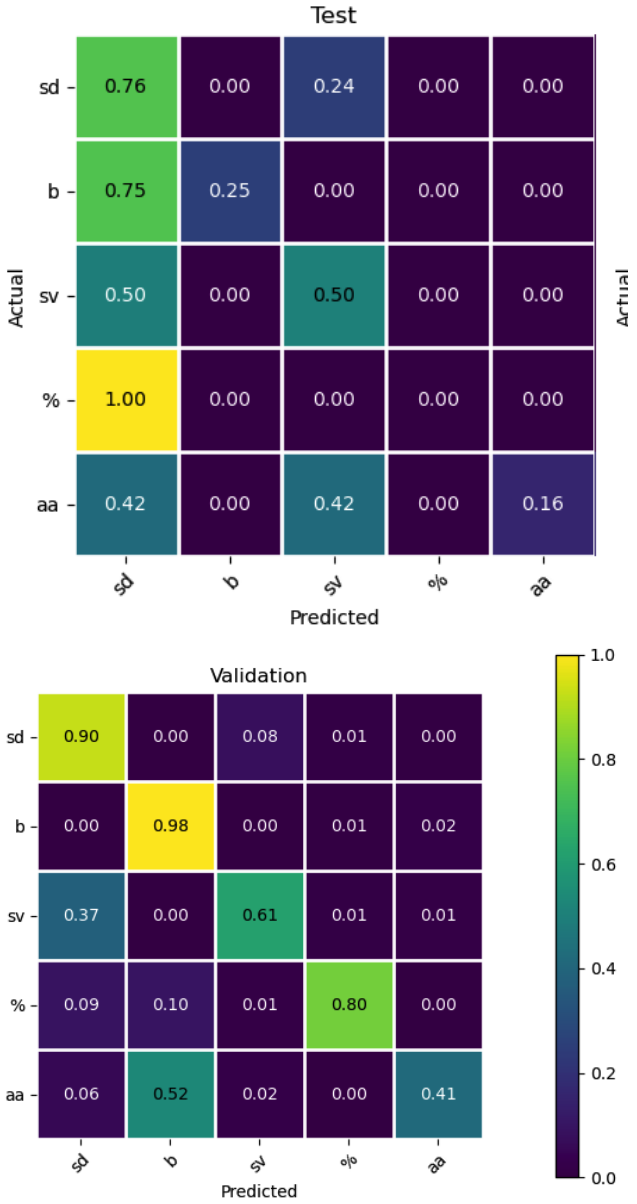


Figure 4.3: Confusion Matrix: Probabilistic representation+LSTM. This illustration only includes classes with the largest support. The classes shown are: sv=statement-opinion, sd=statement non-opinion, aa=agree/accept, b=acknowledge, and %=abandoned.

Table 4.2: Training, validation & test accuracy of all the models

model	acc	val_acc	test_acc(GitHub)
GloVe+LSTM	0.5089	0.5195	0.3714
Prob+LSTM	0.7672	0.7694	0.4412
USE	0.7247	0.6951	0.5071
USE+LSTM	0.3841	0.4257	0.4074
BERT	0.7151	0.7151	0.4063

precision of USE over all tags is 53%. The average recall is 51%, and the average F_1 score is 42%. A difference of only 2% between precision and recall shows that the results of the model are consistent. BERT is one of the newest models for transfer learning; however our results show that fine-tuning BERT doesn't improve performance much in comparison to the Universal Sentence Encoder. Prior work has shown that BERT does not benefits as much from fine-tuning as other embeddings [132].

Identification of Best Performing Model

As a part of our study, we demonstrate a dialogue act classification system for GitHub issue comments. Due to the lack of publicly available training sets of formal teamwork dialogues, we formulated the problem as a transfer learning task, using both sentence-level and word-level embedding models to leverage information from the SwDA dataset. One of the significant contributions of our work is identifying the embedding model that performs best on issue comments. We used GloVe, probabilistic representation, USE, and BERT embedding to train five different models. USE showed the best performance with an accuracy of 50.71%.

Table 4.3: Precision, Recall, & F_1 score of all the tags (USE)

Tag	Precision	Recall	F_1 score	Support
sd	0.51	0.92	0.66	350
b	0.11	0.20	0.14	5
sv	0.47	0.39	0.43	119
%	0.00	0.00	0.00	4
aa	1.00	0.05	0.09	21
ba	0.67	0.21	0.32	29
qy	0.80	0.66	0.72	61
ny	0.00	0.00	0.00	6
fc	0.04	0.50	0.07	2
qw	0.48	0.58	0.52	19
nn	0.00	0.00	0.00	1
bk	0.00	0.00	0.00	0
h	0.00	0.00	0.00	16
qyd̂	0.50	0.17	0.25	6
bh	0.00	0.00	0.00	1
q̂	0.00	0.00	0.00	2
bf	0.00	0.00	0.00	14
fo_o_fw_¨_by_bc	0.00	0.00	0.00	0
na	0.00	0.00	0.00	4
ad	0.67	0.02	0.04	108
2̂	0.00	0.00	0.00	1
bm̂	0.00	0.00	0.00	0
qo	1.00	0.03	0.06	31
qh	0.00	0.00	0.00	0
ĥ	0.00	0.00	0.00	0
ar	0.00	0.00	0.00	1
ng	0.00	0.00	0.00	2
br	0.00	0.00	0.00	0
no	0.00	0.00	0.00	23
fp	1.00	0.83	0.91	6
qrr	0.00	0.00	0.00	0
arp_nd	0.00	0.00	0.00	0
t3	0.00	0.00	0.00	0
oo_co_cc	0.00	0.00	0.00	18
aap_am	0.00	0.00	0.00	2
t1	0.00	0.00	0.00	0
bd	0.00	0.00	0.00	0
ĝ	0.00	0.00	0.00	0
qwd̂	0.00	0.00	0.00	0
fa	1.00	0.14	0.25	7
ft	0.00	0.00	0.00	0
avg / total	0.53	0.51	0.42	859

CHAPTER 5: TEAM PERFORMANCE WITH EMBEDDINGS FROM MULTIPARTY DIALOGUES

This chapter includes contents and figures from the papers titled ”Enayet, A., & Sukthankar, G. (2021, January). Analyzing team performance with embeddings from multiparty dialogues. In 2021 IEEE 15th International Conference on Semantic Computing (ICSC) (pp. 33-39). IEEE.” and ”Enayet, A., & Sukthankar, G. (2021). Learning a Generalizable Model of Team Conflict from Multiparty Dialogues. International Journal of Semantic Computing, 15(04), 441-460.”.

This chapter describes our procedure for computing embeddings using doc2vec [58], an unsupervised method that is used to create a vector representation of the team dialogue. We compare the performance of different possible inputs to doc2vec: 1) dialogue acts, 2) sentiment analysis, and 3) syntactic entrainment.

Table 5.1: Dataset Statistics

Dataset	#Utterances	#Tokens
SwDA	200,052	19,000
Teams Corpus	110,206	573,200

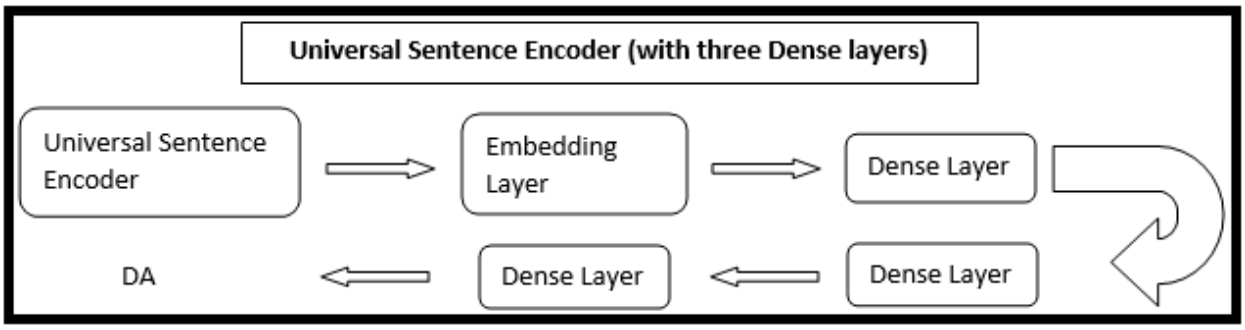


Figure 5.1: Dialogue Act Classifier Architecture.

Dialogue Acts

Dialogue acts can be created from the semantic classification of dialogue at the utterance level to identify the intent of the speaker. A transfer learning approach was used to tag utterances of the Teams corpus using the DAMSL (Discourse Annotation and Markup System of Labeling) tagset. Figure 5.1 shows the architecture of our dialogue act classifier, which was constructed using the Universal Sentence Encoder; we selected USE for its ability to achieve consistently good performance across multiple NLP tasks [14]. There are two different variants of the model: 1) a transformer architecture, which exhibits high accuracy at the cost of increased resource consumption and 2) a deep averaging network that requires few resources and makes small compromises for efficiency. The former uses attention-based, context-aware encoding subgraphs of the transfer architecture. The model outputs a 512-dimensional vector. The deep averaging network works by averaging words and bigram embeddings to use as an input to a deep neural network. The models are trained on web news, Wikipedia, web question-answer pages, discussion forums, and the Stanford Natural Language Inference (SNLI) corpus, and are freely available on TF Hub.

We selected the USE Transformer-based Architecture model with three dense layers and a softmax activation function. Figure 5.1 shows the architecture of our DA classification model, which achieves a validation accuracy of 70%.

The model was fine-tuned using the Switchboard Dialogue Act Corpus (SwDA) dataset. SwDA is one of the most popular public datasets for DA classification. It consists of 1155 human-to-human telephone speech conversations, tagged using 42 tags from the DAMSL tagset. Table 5.1 shows the statistics of both SwDA and the Teams corpus.

Table 3.6 shows examples from the SwDA training dataset, and Table 3.1 shows examples from Teams corpus. Each team dialogue generates a unique sequence where each element of the se-

quence represents the dialogue act of the corresponding utterance. This sequence of dialogue acts is then used as an input to doc2vec algorithm to create the embedding.

Sentiment Analysis

Another option is to represent the team dialogue as a series of changes in the emotional state of the team. This can be done by applying sentiment analysis to the individual utterances. Sentiment analysis is the task of predicting the emotion or attitude of the speaker; we are using the TextBlob python implementation [114] to determine sentiment polarity of each utterance in the dialogue. The polarities are float values which lies between -1 and 1 representing negative, positive and neutral sentiment. For each team the unique sequence of these polarities is used as input to doc2vec, where each element of the sequence represents the polarity of the corresponding utterance. This representation encodes transitions in the emotional state of the team across the duration of the task.

Entrainment

Entrainment is one form of linguistic coordination in which team members adopt similar speaking styles during conversation. Here we evaluate the performance of a syntactic entrainment embedding based on Rahmi and Litman's [89]'s work that encodes the propensity of subsequent speakers to make similar lexical choices. Eight lexical categories were used: noun (NN), adjective (JJ), verb (VB), adverb (RB), coordinating conjunction (CC), cardinal digit (CD), preposition/subordinating conjunction (IN), and personal pronoun (PRP) . To calculate the entrainment between two speakers we follow the method proposed by Danescu et al. [19] shown in Equation 5.1. $Ent_c(x, y)$ is the entrainment of speaker y to speaker x, c is the lexical category, e_{yx^c} represents the event where speaker y utterance immediately follows the speaker x utterance and contains c, e_x^c is the event

when utterance (spoken to y) of speaker x contains c.

$$Ent_c(x, y) = p(\frac{e_{yx^c}}{e_x^c}) - p(e_{yx^c}) \tag{5.1}$$

The NLTK part-of-speech (POS) tagger was used to tag all the utterances with their respective lexical categories. A directed weighted graph was generated for each dialogue linking speakers with positive entrainment. The structure of this graph encodes the entrainment relationships between team members. To translate the graph into a feature representation, six graph centrality kernel functions were applied to represent each node of the team graph. Table 5.2 describes the kernel functions: (1) PageRank (2) betweenness centrality (3) closeness centrality (4) degree centrality (5) in degree centrality (6) Katz centrality. To create the final team representation, the vectors of individual nodes were averaged, and doc2vec was applied to create the embedding. With eight lexical categories and six kernel functions, the length of the feature vector is 48. This method corresponds to the Kernel version of Entrainment2Vec [89] and achieves comparable performance when applied to the whole dialogue.

Our implementation is slightly different from that of [89] and [19] in two aspects. First, we are using the NLTK POS tagger to assign lexical categories to the utterances instead of using LIWC-derived categories. Second, we are using six graph kernel algorithms instead of ten. The POS tagging reflects the sentence's syntactic structure; we have carefully selected the POS categories that are consistent with the conventional English part of speech categories used by [89] and [19]. While calculating the entrainment, we do not consider the actual word and its context; therefore, this embedding only captures syntactic features, not semantics.

Table 5.2: Entrainment Kernel Functions

Kernel Function	Description
PageRank	Ranks the node based on the quality and number of incoming links
Betweenness centrality	Measures the centrality of the node based on the shortest paths (measures information flow)
Closeness centrality	Reciprocal of the sum of the length of the shortest paths between the node and the rest of the graph (measures efficiency of information spread)
Degree centrality	Number of incoming and outgoing entrainment connections
In-degree centrality	Number of incoming entrainment connections only
Katz centrality	Measures the number of walks between two nodes, reflecting its relative influence on neighbors.

Doc2vec

Le and Mikolov [58] introduced doc2vec as an unsupervised learning algorithm to generate distributed vector representations of text of arbitrary size; it is inspired by the word2vec model [70]. They proposed two different models for learning numerical representations of text: 1) Distributed Memory Model of Paragraph Vectors (PV-DM) 2) paragraph vector with a distributed bag of words (PV-DBOW).

Distributed Memory Model of Paragraph Vectors (PV-DM) uses both word vectors and paragraph vectors to predict the next word. It attempts to learn paragraph vectors that can predict the word given different contexts sampled from the text. The context size is a tuneable parameter, and a sliding window of arbitrary context size generates multiple context samples. Doc2vec works by averaging these word vectors and paragraph vectors to predict the next word. It employs stochastic

gradient descent to learn word and paragraph vectors. The resultant paragraph vectors serve as a feature vector of the corresponding paragraph and can be used as an input to machine learning models like SVM and logistic regression.

Paragraph vector with a distributed bag of words (PV-DBOW) ignores the context words and attempts to predict randomly selected words from the paragraph. At each iteration of stochastic gradient descent, it classifies a randomly selected word from the sampled text window using paragraph vectors.

Instead of using doc2vec on the raw team dialogues, doc2vec was applied to the output of the dialogue act classifier, sentiment analysis, and syntactic entrainment. This procedure enables us to disentangle the contribution of different elements of team communication at predicting conflict.

Datasets

This analysis includes results from three datasets: 1) multiplayer cooperative board games (Teams corpus) [61]; 2) software engineering teams (GitHub issue comments); and 3) military team communications [48]. We test the generalizability of the embeddings learned on the Teams corpus on the two datasets collected from software engineering and military teams. The Teams corpus is the most complete dataset since it is the only one that contains post-task process conflict ratings.

We initially apply our proposed methodology on the Teams corpus dataset collected by Litman et al. [61]. We are using process conflict z-scores to represent team conflict. Jehn et al. have identified that low process conflict scores indicate good team performance and vice versa [44]. To study the problem of early prediction of team conflict, we divide each dialogue into three equal sections that correspond to the knowledge-building, problem solving, and culmination teamwork phases. Our final classification dataset consists of 12 patterns per dialogue, which are generated

from applying the three methods (semantic, sentiment, syntactic) to the whole time period, as well as the initial, middle and final segments.

We test the generalizability of the learned models on software engineering teams (GitHub issue comments) and military team communications.

Experimental Setup

Teams were divided into low and high conflict teams based on their process conflict z-scores, and classification accuracy was measured. Doc2vec was used to generate the vector representation of all the patterns. Doc2vec comes in two different flavors: 1) Distributed Memory Model of Paragraph Vectors (PV-DM) and 2) Distributed Bag of Words version of Paragraph Vector (PV-DBOW). Table 5.3 shows the comparison of PV-DM & PV-DBOW when applied to the complete dialogue. The main difference between PV-DM and PV-DBOW is, unlike PV-DBOW, PV-DM keeps track of the context while encoding. The high performance of PV-DM on DAs and sentiment patterns, compared to PV-DBOW, confirm that the sequences contain meaningful information. Our results show that the performance difference of the PV-DM and PV-DBOW using the dialogue act and sentiment embeddings are statistically significant ($p < .01$). The difference of the PV-DM and PV-DBOW using the entrainment embedding is not statistically significant (p=0.754). PV-DM gives consistent performance across all the three features sets, making it a better candidate for detailed analysis. Through extensive experiments, we identified that PV-DM with epoch size of 5, negative sampling 5, and window size 10 works best for our setting. By default, we only report results for PV-DM.

We evaluated the performance of both logistic regression and the support vector machine (SVM) classifier on the full dialogue (shown in Table 5.4). SVM clearly performed better than logistic re-

Table 5.3: Doc2Vec Comparison

	PV-DBOW		PV-DM	
	Accuracy	F1-Score	Accuracy	F1-Score
Dialogue Act	57.89	58.25	**68.42**	**68.77**
Sentiment	55.26	55.48	**78.94**	**77.53**
Entrainment	55.26	55.04	**60.52**	**60.77**

Table 5.4: Comparison of Supervised Classifiers

	Logistic Regression		SVM	
	Accuracy	F1-Score	Accuracy	F1-Score
Dialogue Act	63.15	63.15	**68.42**	**68.77**
Sentiment	71.05	70.86	**78.94**	**77.53**
Entrainment	**63.15**	**63.15**	60.52	60.77

gression using the dialogue act and sentiment embeddings. Logistic regression seemed to perform better on entrainment compared to the SVM. We report the detailed comparison of the two classifiers by incrementally increasing the length of the dialogues in Section 6. To remain consistent with the previous work [89], SVM was used for the teamwork phase comparison.

CHAPTER 6: EXPERIMENTAL EVALUATION OF HYPOTHESIS H1-H4

Results on Teams Corpus

This chapter includes contents and figures from the papers titled ”Enayet, A., & Sukthankar, G. (2021, January). Analyzing team performance with embeddings from multiparty dialogues. In 2021 IEEE 15th International Conference on Semantic Computing (ICSC) (pp. 33-39). IEEE.” and ”Enayet, A., & Sukthankar, G. (2021). Learning a Generalizable Model of Team Conflict from Multiparty Dialogues. International Journal of Semantic Computing, 15(04), 441-460.”.

Table 6.1 presents the classification accuracy of the three embeddings on the whole dialogue. SVM exhibits the best classification accuracy of 78.94% on sentiment based vectors, followed by dialogue act based vectors. Figure 6.1 visually illustrates the effects of different embeddings. By plotting the vectors in 2d using t-Distributed Stochastic Neighbor Embedding (TSNE), we can observe the formation of two clusters, representing teams with high social outcomes and low social outcomes in the dialogue act and sentiment vectors, whereas the entrainment ones are intermixed.

Table 6.1 shows the accuracy of the conflict classifier across the duration of the games. The sentiment classifier achieved the best accuracy when the whole dialogue was used and exhibited consistent performance across all team phases. The dialogue act embedding was the best at the

Table 6.1: Accuracy by Team Phase

Phase	Dialogue Act		Sentiment		Entrainmemt	
	Accuracy	F1-Score	Accuracy	F1-Score	Accuracy	F1-Score
Whole	68.42	68.77	**78.94**	**77.53**	60.52	60.77
Initial	**71.05**	**71.35**	65.78	62.84	42.10	42.42
Middle	**73.68**	**73.31**	65.78	59.18	47.36	46.78
End	68.42	68.68	**71.05**	**71.19**	60.52	60.32

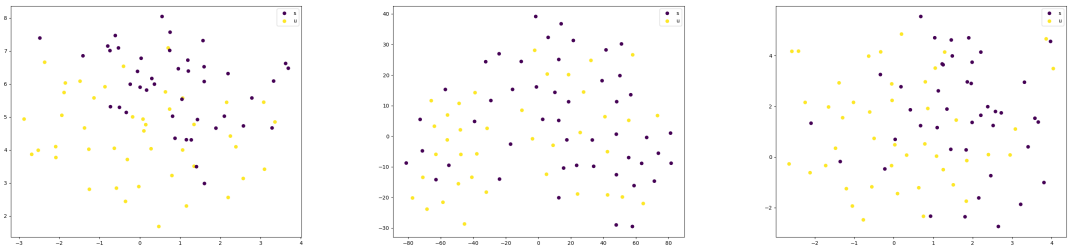


Figure 6.1: t-SNE representation of vectors in 2D, where 'S' represents the teams with low process conflict scores and 'U' represents the teams with high process conflict scores. Both sentiment (left) and dialogue act embedding (right) show a better class separation than entrainment (center). Note that the axes have no explicit meaning.

initial phase, making it a good choice for the "thin-slice" problem of rapidly diagnosing teamwork health from a small sample of utterances. Syntactic entrainment lagged behind the sentiment and semantic analysis, but performance improved during the final phase. Note that each phase was analyzed separately, rather than cumulatively.

For statistical testing, we generated 30 results for each phase using each embedding. Since some of the result distributions (Figure 6.2) failed the D'Agostino-Pearson normality test, the Kolmogorov-Smirnov test was used for significance testing. The performance differences between each pair of embeddings were statistically significant ($p < 0.01$). However the differences between the initial and end phase results for the sentiment and entrainment embeddings were not significant (Table 6.2). Semantic and sentiment based vectors outperformed the syntactic entrainment vectors at the classification task across all phases.

Preliminary results (Table 5.4) showed that entrainment vectors perform slightly better when used with logistic regression than with SVM. To further analyze the finding and test our third hypotheses (**H3**), we check the embeddings' performance as the dialogue progresses. For this purpose, we divide the dialogues into 20 phases. Starting from the first phase of the dialogue, we incrementally

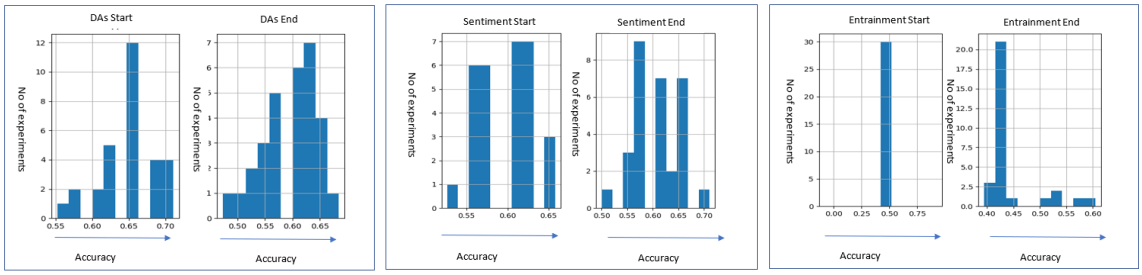


Figure 6.2: Distribution of embedding results for initial and final teamwork phases for dialogue acts (left), sentiment (middle) and entrainment (right)

Table 6.2: Comparison of approaches during the initial (knowledge discovery) and culmination (final) phases

	Knowledge Discovery		Culmination		
	min	max	min	max	p-value
Dialogue Act	0.552632	0.710526	0.473684	0.684211	**2.48e-05**
Sentiment	0.526316	0.657895	0.500000	0.710526	0.455695
Entrainment	0.4210	0.4210	0.394737	0.605263	0.594071

increase the dialogue's length by adding the next phase into it. This is different from testing on knowledge building phase, problem-solving phase, and culmination phase, where while training and testing on any specific phase, we did not include utterances from previous phases. Figure 6.3 shows the trend of classification performance of the different embeddings with both logistic regression and SVM classifiers as the dialogue progresses. Results showed that both sentiment and dialogue act embeddings dominant across the whole timeframe. The results also reject **H3** by showing that the entrainment performance does not improve as the dialogue progresses.

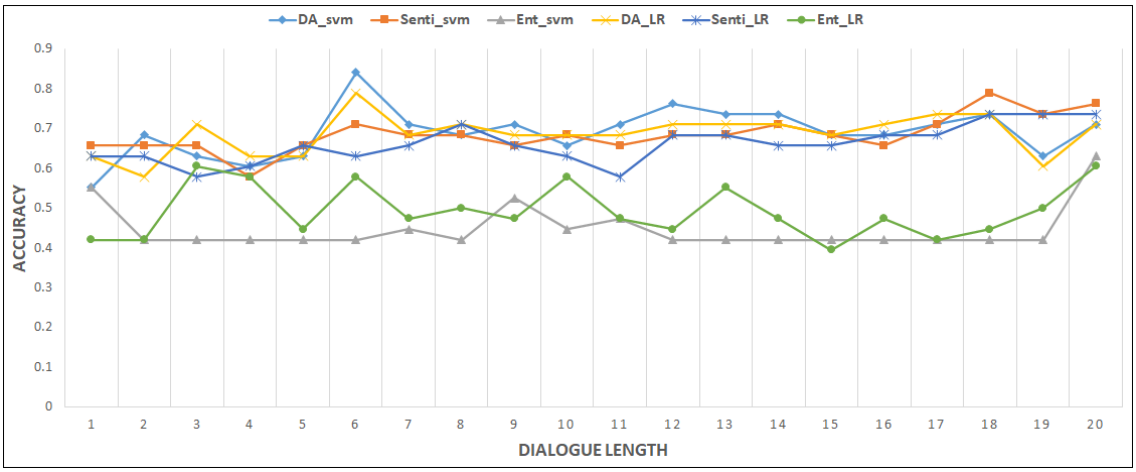


Figure 6.3: Conflict prediction accuracy of different embeddings on the Teams corpus as the dialogue progresses. The classifiers (SVM and logistic regression) using the entrainment embedding (green and gray lines) perform consistently worse across the whole dialogue.

Results on Dataset Generalization

One of our research goals is to create team communication embeddings that generalize well across datasets, since there are few team communication datasets and some of them are extremely small. To evaluate generalizability, we apply our pre-trained models (dialogue acts, sentiment, entrainment) on the GitHub issue comments and military dialogues.

Table 6.3 shows the performance of the different embeddings on the GitHub issue comments. The dialogue act embedding outperforms the other ones under both classifiers and achieves a comparable performance to the original dataset. The dialogue act embedding also outperforms sentiment on the small military team communication dataset (Table 6.4). Unfortunately, the pre-trained entrainment embedding completely failed on this problem. One issue with the military dataset is that it features significantly larger teams (15 members on average) than the Teams corpus (3-4 members). We believe that graph based entrainment measures do not generalize well across larger

Table 6.3: Performance on GitHub Issue Comments Dataset

	Logistic Regression	SVM
Dialogue Act	**66.00**	**68.00**
Sentiment	58.00	60.00
Entrainment	42.00	42.00

Table 6.4: Performance on Military Teams Dataset

	Logistic Regression	SVM
Dialogue Act	**100.00**	**100.00**
Sentiment	90.00	60.00
Entrainment	-	-

graphs since the kernel measures are very dependent on graph size. Also the length of dialogues in GitHub issue comments is short compared to the Teams corpus; many team members only have one utterance in a dialogue. The small number of utterances from a team member doesn't facilitate effective computation of entrainment.

Improving Conflict Detection Performance

Our long-term goal is to create a proactive assistant agent that can rapidly detect team conflicts using the dialogue act embedding. To do this, we want to maximize the F1-score of the high conflict class (unsuccessful teams). Fine-tuning the model on the target dataset is one way to improve the pre-trained model's performance on the target dataset. Due to the small size of the Teams corpus, we do not use an extensive deep learning model; to analyze the performance of the classifier when samples from the target dataset are used for training along with the actual training corpus, we add five high conflict dialogues from the GitHub issue comments dataset to the training dataset. This is not possible to do with the military dataset which lacks good examples of conflict. We have

intentionally selected a minimal number of samples from the target dataset to avoid cheating the generalizability check. Table 6.5 shows the comparison of F1-scores of the individual classes when the dialogue act embedding is trained with and without supplemental high conflict examples. The GitHub dataset contains 50 dialogues, of which we are using 5 dialogues for training and 45 for testing. Incorporating GitHub high conflict samples in training the dialogue act embedding also improved the accuracy of the low conflict class. We were able to use a similar approach to boost the performance of the sentiment embedding, but the final performance remained lower than the dialogue act embedding.

Table 6.5: Performance on GitHub Issues Dataset With vs. Without High Conflict Training Examples

SVM Classifier			
	Accuracy (overall)	Low Conflict (F1-Score)	High Conflict (F1-Score)
Without	68.88	80.00	30.00
With	**73.33**	**81.00**	**54.00**
Logistic Regression Classifier			
	Accuracy (overall)	Low Conflict (F1-Score)	High Conflict (F1-Score)
Without	71.11	81.00	38.00
With	**75.55**	**84.00**	**52.00**

Conclusion

This study presents an evaluation of different embeddings for predicting team conflict from multiparty dialogue. Embeddings were extracted from three types of features: 1) dialogue acts 2) sentiment polarity 3) syntactic entrainment. Results confirm the effectiveness of both sentiment (**H2**) and dialogue acts (**H1**). However, experiments failed to confirm that classification based on syntactic entrainment signficantly improves over time (**H3**). Although there are many other ways to measure linguistic synchronizaton, it seems less promising for integration into an agent assistance

system. The dialogue act embedding is strong during the initial phase making it a good candidate for diagnosing the health of team formation activity. A continuous team monitoring agent assistant system might do better with sentiment analysis, assuming training data availability.

The highly specialized nature of the team communication produced by software engineering and military teams make them excellent candidates to evaluate the learned embeddings. We test models trained on the Teams corpus on these other datasets. The dialogue act embedding generalized better than sentiment and entrainment on real-world datasets from software engineers and military teams (**H4**). Results show that fine-tuning on the target dataset improves performance. Sentiment embeddings show some potential but seem more promising when trained and tested on the same corpus. Due to its usage of graph kernels, the entrainment feature vector is highly dependent on consistent team sizes and did not generalize well on either corpus.

CHAPTER 7: AN ANALYSIS OF DIALOGUE ACT SEQUENCE SIMILARITY ACROSS MULTIPLE DOMAINS (H5)

This chapter includes contents and figures from the paper titled ”Enayet, A., & Sukthankar, G. (2022, June). An Analysis of Dialogue Act Sequence Similarity Across Multiple Domains. In Proceedings of the Thirteenth Language Resources and Evaluation Conference (pp. 3122-3130).”.

As a part of this dissertation, we also analyse how dialogue act sequences vary across different domains in order to anticipate the potential degradation in the performance of learned models during domain adaptation. This chapter presents the detail of our proposed method and the findings.

Introduction

Transfer learning is commonly used in natural language processing to compensate for paucity of data; a machine learning model can often be trained on a single large source dataset and then fine-tuned for smaller target datasets. Unfortunately many machine learning models perform poorly when exposed to *domain shifts*, distributional differences between source and target datasets. Studies have shown that, unlike machine learning algorithms, humans are more robust to these natural distribution shifts [73].

In this dissertation, we focus on the problem of learning models for discourse analysis that generalize across different communication settings. Discourse is often represented as a sequence of dialogue acts (DAs) where each DA represents the functional purpose of the utterance in the conversation (e.g., statement, question, agreement). Dialogue modeling systems not only analyze the

content of the utterance, but also the context of neighboring dialogue acts to track conversational state; for instance, agreement dialogue acts often follow questions. Due to differences in the linguistic features of training and test data, natural distribution shifts may occur [53]. In dialogue models that rely on the context of utterances, we hypothesize that differences in DA patterns will affect model performance.

This study presents a methodology for predicting the potential degradation in the performance of learned models during domain adaptation. Our analysis shows that dialogue sequences from related domains possess similar n-gram frequency distributions. This similarity can be quantified by measuring the average Hamming distance between subsequences drawn from different datasets. We analyze the similarity of the dialogue acts across eight different datasets: SwDA, AMI (DialSum), GitHub, Hate Speech, Teams, Diplomacy Betrayal, SAMsum, and Military (Army). These datasets represent many types of discourse including collaboration, formal discussion, strategic planning, and social media exchanges. Rather than evaluating performance on a specific dialogue modeling task, we evaluate the suitability of embeddings learned from DA sequences for discriminating between discourse from different datasets. Our experiments demonstrate that when dialogue acts sequences from two datasets are dissimilar they lie further away in embedding space, making it possible to train a classifier that is robust to data perturbations, such as random deletion and tag swapping. Our objective is to provide intuition on the transferability of learned models that utilize dialogue act patterns to make predictions; our research findings have implications for many critical applications including conversational agents, question answering systems, role identification, and speech recognition.

Related Work

Dialogue act sequences have been leveraged for a variety of NLP tasks such as coreference resolution [2], misunderstanding detection [1], abstractive summarization [35], discourse chunking [69], information need classification [33], and conversational models [54]. Example applications include situational-based dialogue management systems [59], agenda-based simulators for training dialogue managers [102], semi-automated negotiation [134], and dynamic dialogue selection [95].

Dialogue act classifiers tag each utterance with a label according to a taxonomy of conversational functions. Many dialogue act classification techniques make use of the labels of the surrounding utterances such as the Contextual Dialogue Act classifier (CDAC) [3], n-gram models [126, 36], and unsupervised multimodal feature-based techniques [30]. Neural architectures [116] are commonly employed for dialogue act classification including the dual-attention hierarchical RNN [60] and generative models [115].

However, there is little work on the problem of measuring similarity between two dialogues. Lavi et al. (2021) introduce a method ConvEd to calculate the similarity between two conversations to support the retrieval of relevant customer service interactions for chatbots. ConvEd measures the edit distance between the two conversations by counting the insertion, deletion, and substitution operations required to align the two conversations. Unlike our work, ConvEd measures similarity by calculating an embedding over the original utterances, rather than the dialogue act tags.

Researchers have developed techniques for efficient computation of document similarity [26], node similarity [92], entity resolution [16], and query expansion [62], Many of the proposed approaches exploit word embeddings for the computation of similarity [26, 92, 15, 62]. We use Doc2Vec [58], a variant of Word2Vec [70], since we are interested in document level (dialogue) embeddings rather than word level. The Distributed Memory Model of Paragraph Vectors (PV-DM) model

of Doc2Vec generates embeddings by sampling context windows of user-defined sizes from a paragraph and preserving the most meaningful information contained in the sequences present in those context windows. The next section describes our methodology for quantifying the similarity of dialogue act sequences.

Methodology

A DA classifier was used to extract sequences of dialogue acts from sets of dialogues. Our analysis was performed on eight datasets that span a rich cross-section of human social interactions. In the following sections, First we present the frequency distribution of the dialogue act n-grams. Then we introduce our proposed similarity measure for predicting generalizability performance: the percentage of zero Hamming distance subsequences of fixed window size drawn from different datasets.

We contrast this method to one of the most commonly used methods of calculating document similarity, a Doc2Vec embedding. This type of embedding is often used as a basis for other dialogue modeling tasks. We measure the cosine similarity of discourse using the embeddings obtained through Doc2Vec. Then we study how effective the embedding is at discriminating between dialogue instances drawn from different datasets, using a discriminative distance method. Binary classifiers are trained to classify the dataset from a DA sequence represented in the Doc2Vec embedding; using the learned models, we identify the most confusing pairs of datasets for a binary classifier. We show that the most confusing datasets are typically collected within the same communication context and are highly similar according to both the dialogue act n-gram and Hamming distance analysis. These confusing pairs are strong candidates to be compatible domain adaptation source and target tasks. We have made the dataset of dialogue act sequences collected from different communications settings available at `https://github.com/`

ayeshaEnayet/DAC-USE (under DomainShift).

Dialogue Act Classification

First we apply our Universal Sentence Encoder (USE) based DA classification model, trained on the SwDA dataset, to tag the utterances of all the datasets. We use the SwDA-DAMSL tagset available at https://web.stanford.edu/~jurafsky/ws97/manual.august1.html. USE is itself trained on a variety of datasets, including discussion forums, and it exhibits a good performance on a variety of NLP tasks [14]. The code and details for the DA classification model are available at https://github.com/ayeshaEnayet/DAC-USE. We selected the USE based model due to its ability to generalize effectively across dialogue (discussion) datasets. The test accuracy of our classification model is 72%, and validation accuracy is 70% which is comparable to most of the DA classification approaches. The DA classifier does not consider surrounding utterances to predict the tag of the current utterance; classification is performed solely on the basis of the information present in the embedding of a single utterance.

The DA classifier takes a sequence of utterances as its input and returns the sequence of DAs, where each DA corresponds to one utterance. Table 7.1 shows the top three most frequent unigrams, bigrams, trigrams, 4grams, and 5grams of the datasets used in this analysis. There is some overlap in the DA n-grams across all datasets; for instance sequences of sd (statement-non-opinion) are common across all datasets.

Datasets

Datasets were selected to represent a cross-section of communication domains including social media exchanges, collaboration, formal discussion, telephonic conversation, and strategic dialogues.

Table 7.1: N-gram frequency distribution: top three most frequent unigrams, bigrams, trigrams, 4grams, 5grams of all the datasets. Sequences of sd (statement-nonopinion) are common across all datasets. The most frequent tags in this table are sd: Statement-non-opinion, b: Acknowledge, %: Uninterpretable, sv: Statement-opinion, ad: Action-directive, qy: Yes-No-Question, fc: Conventional-closing, qh: Rhetorical-Questions.

Dataset	**Unigrams**	**Bigrams**	**Trigrams**	**4grams**	**5grams**
Teams	(sd),(b),(%)	(sd,sd),(sd,b),(b,sd)	(sd,sd,sd), (sd,sd,b), (sd,b,sd)	(sd,sd,sd, sd), (sd,sd,sd,sd), (sd,sd,sd,b)	(sd,sd,sd,sd, sd), (sd,sd,sd,sd,b), (sd, sd,sd,b,sd)
GitHub	(sd),(sv),(ad)	(sd,sd),(sd,sv),(sv,sd)	(sd,sd,sd), (sv,sd,sd), (sd,sd,ad)	(sd, sd, sd, sd), (sd, sd, sd, ad), (sv, sd, sd, sd)	(sd, sd, sd, sd, sd), (sd, sd, sd, sd, ad), (sd, sv, sd, sd, sd)
Army	(sd),(qy),(%)	(sd,sd),(qy,sd),(sd,qy)	(sd,sd,sd), (sd,sd,qy), (qy,sd,sd)	(sd, sd, sd, sd), (sd, sd, sd, qy), (qy, sd, sd, sd)	(sd, sd, sd, sd, sd), (qy, sd, sd, sd, sd), (sd, sd, sd, sd, qy),
SAMsum	(sd),(sv),(fc)	(sd,sd),(sv,sd),(sd,sv)	(sd,sd,sd), (sd,sv,sd), (sv,sd,sd)	(sd, sd, sd, sd), (sd, sd, sv, sd), (sd, sv, sd, sd)	(sd, sd, sd, sd, sd), (sd, sd, sd, sv, sd), (sd, sd, sv, sd, sd)
Hate Speech	(sd),(sv),(fc)	(sd,sd),(sv,sd),(sd,sv)	(sd,sd,sd), (sd,sv,sd), (sv,sd,sd)	(sd, sd, sd, sd), (sd, sd, sv, sd), (sd, sv, sd, sd)	(sd, sd, sd, sd, sd), (sd, sd, sd, sd, qh), (sd, sd, sd, qh, sd)
SwDA	(sd) (sv)(b)	(sd, sd),(sd, b),(b, sd)	(sd, sd, sd), (sd, sd, b), (sd, b, sd)	(sd, sd, sd, sd), (sd, sd, sd, b), (sd, sd, b, sd)	(sd, sd, sd, sd, sd), (sd, sd, sd, b, sd), (sd, sd, sd, sd, b)
AMI	(sd), (b),(sv)	(sd,sd),(b,sd),(sv, sd)	(sd,sd,sd), (b, sd,sd), (sd, sv, sd)	(sd, sd, sd, sd), (b, sd, sd, sd), (sd, sv, sd, sd)	(sd, sd, sd, sd, sd), (b, sd, sd, sd, sd), (sd, sd, sd, sd, sv)
Diplomacy	(sd),(sv),(qy)	(sd,sd),(sv,sd),(sd,sv)	(sd,sd,sd), (sd,sd,sv), (sv,sd,sd)	(sd, sd, sd, sd), (sv, sd, sd, sd), (sd, sd, sd, sv)	(sd, sd, sd, sd, sd), (sd, sv, sd, sd, sd), (sd, sd, sd, sd, sv)

Some of these datasets are quite large, but many are too small to support complex machine learning models. Our analysis was performed on a balanced dataset with 50 randomly sampled dialogues selected from each dataset, except for the Military dataset which only has 22 examples. A noisy version of this dataset was also created by randomly deleting and swapping dialogue act labels (see Section 7 for details). All the datasets contain dialogue in the English language. We perform analysis on SwDA, AMI (DialSum), GitHub, Hate Speech, Teams, Diplomacy Betrayal, SAMsum, and Military (Army).

Sequence Similarity

The Hamming distance between two sequences is the number of positions where the sequences have different values. We extract all the possible subsequences of lengths four and five from the output of the DA classifier and calculate the Hamming distance between the sequences from all the datasets. To score each sequence, we increment the count by one for every pair of subsequences possessing a Hamming distance of zero. The similarity score between two dialogues is represented as a percentage. The final similarity score between datasets is quantified by taking the average of the scores.

Embeddings

Most machine learning models start by learning a lower dimensional representation of the data that can be used by the NLP pipeline. Each discourse is initially represented as a sequence of dialogue acts. Sequences of DAs are treated as documents, with the DAs forming the vocabulary of the document. We apply Doc2Vec [58], a technique to learn paragraph vectors, to learn embeddings from these sequences of dialogue acts. The Distributed Memory (DM) model of the Doc2Vec was used because of its ability to generate embeddings by considering the context window of varying sizes, as opposed to Distributed Bag of Word (DBOW) model, which does not consider the context when learning embeddings. Our analysis was performed with the Doc2Vec function from the Gensim library. We use PV-DM with epoch size of 5, negative sampling 5, and window size 5. We then apply both the discriminative distance method and cosine similarity measures to the embeddings.

Discriminative Distance: Discriminative distance was used to identify the most confusing dataset pairs. We train a support vector machine (SVM) binary classifier on the embeddings learned from

Doc2Vec; its aim is simply to identify the dataset. The most confusing pairs are the ones that have similar embeddings. If the classifier exhibits a high accuracy, it means that the embedded representation is sufficiently distinct to allow the classifier differentiate between the two datasets. We evaluated the SVM with both a linear and non linear (radial basis function) kernel.

Cosine Similarity: Cosine similarity is a measure of similarity between two vectors calculated by taking the cosine of the angles between two embeddings. We measure the cosine similarity between the embeddings of all the datasets that we obtain through Doc2Vec.

Experimental Analysis

The datasets can be grouped by communication setting, with some datasets falling into multiple categories. The Teams, GitHub, and Army datasets are collaborative dialogues gathered from team communications. The SAMsum and Hate Speech datasets are social media exchanges. The Diplomacy and Teams datasets were collected from game communication. GitHub also falls under the social media category, but the dialogues in this dataset are more formal and goal-oriented than SAMsum and Hate Speech. SwDA is a telephonic communication dataset composed of non-goal-oriented discussion between two people. Diplomacy and Army are both good examples of strategic planning. The AMI meeting and GitHub datasets are goal-oriented formal discussion. Table 7.2 provides an overview of our categorization.

Table 7.1 shows the result of our n-gram frequency distribution analysis and gives the top three most frequent unigrams, bigrams, trigrams, 4grams, and 5grams of all the datasets. The most frequent unigram, bigram, trigram, and 4gram in social media dialogues like SAMsum and Hate Speech are the same. Also, the Yes-No question (qy) is one of the major categories in strategic dialogues. The SwDA and AMI both have statement (sd), opinion (sv), and acknowledgment (b) as

Table 7.2: Categorization of datasets.

Datasets	**Category**
Teams, GitHub, Army	Collaboration
SAMsum, Hate Speech, GitHub	Social Media
SwDA	Discussion (informal/non-goal-oriented)
Diplomacy, Army	Strategic planning
Diplomacy, Teams	Gameplay
AMI, GitHub	Discussion (formal/goal-oriented)

frequently occurring categories in the discourse. Uninterpretable (%) is a prominent unigram in social media datasets. GitHub and Diplomacy datasets have bigram sequences in common; this may occur in both datasets because members propose solutions to each other. Statement-non-opinion (sd) and Statement-opinion (sv) are the most frequently occurring tags of formal dialogues (AMI and GitHub). In addition to sv and sd, the most prominent unigram in GitHub is Action-directive (ad) because, in these dialogues, the members suggest a course of actions to the other members to solve problems. Similarly, in AMI corpus Acknowledge (b) is one of the most prominent tags.

Figure 7.1 shows the distribution of embeddings of all the datasets on a 2D plane. The distribution indicates that SwDA and Teams are clustered separately from other datasets and have unique embeddings. On the other hand, the SAMsum, Hate Speech, and GitHub dataset embeddings (all from the Social Media category) are intermixed and cover a large area. Social media dialogues tend to have a similar dialogue flow. Diplomacy slightly overlaps with GitHub and is near the Military dataset.

Figure 7.2 shows the classification accuracy of the SVM (with linear kernel) at distinguishing

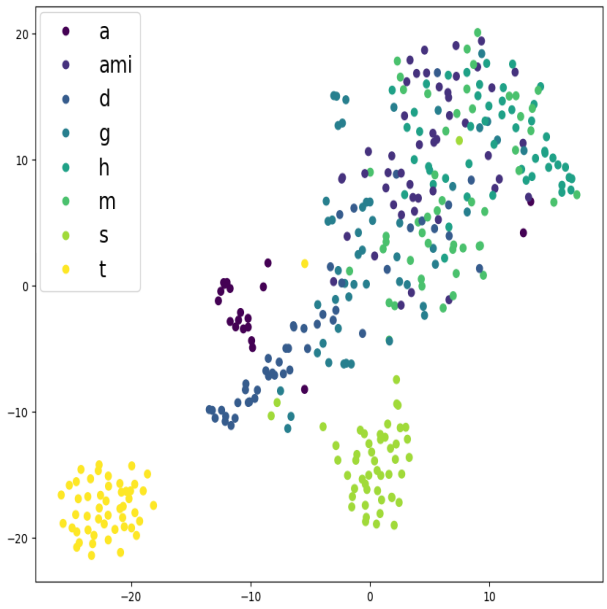


Figure 7.1: Projection of embeddings of datasets in 2D space. ami: AMI, g: GitHub, d: Diplomacy, t:Teams, h: Hate Speech, s: SwDA, m: SAMsum, a: Military (Army).

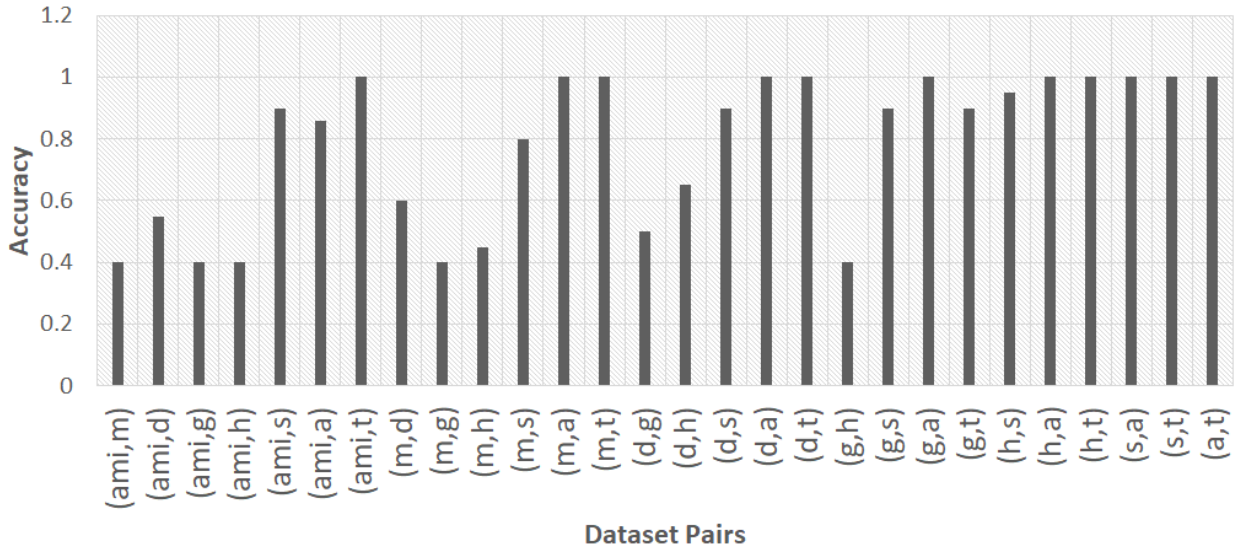
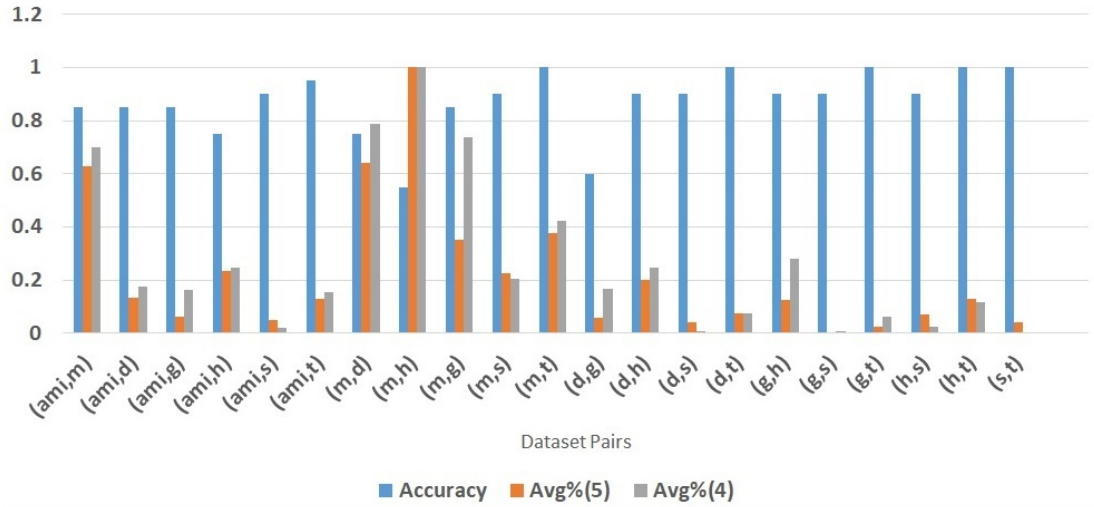


Figure 7.2: Pairwise classification accuracy using SVM with linear kernel and the Doc2Vec embedding. The classification task is simply to identify the dataset. ami: AMI, g: GitHub, d: Diplomacy, t:Teams, h: Hate Speech, s: SwDA, m: SAMsum, a: Military (Army).

between dialogue act sequences drawn from different datasets. Instances are represented using the embedding illustrated in Figure 7.1. This shows that SwDA, Military (Army), and Teams are linearly separable from almost all the datasets and exhibit a high classification accuracy. AMI, Hate Speech, GitHub, and SAMsum have high error rates. On the other hand, Diplomacy lies in between highly separable and inseparable datasets. AMI and GitHub, i.e., the formal discussion datasets, showed a significant overlap with four out of seven datasets. The results also indicate that even dialogues within the same domain may exhibit different communication patterns. The Military and Teams dataset belong to multiple categories but have distinct communication patterns from other datasets.

We validate our ML-based models against the non-ML-based similarity measures. Figure 7.3 shows the comparison of average percentage similarity between pairs of datasets, calculated using Hamming distance, and binary classification accuracy, using the RBF kernel function. The results show that a high similarity between two datasets leads to low binary classification accuracy. SwDA is one of the standard datasets used for the DA classification task. Yet our results show that SwDA is very different from other datasets, as can be observed in Figure 7.1, and gives the highest binary classification accuracy when classified against other datasets. SAMsum showed the lowest binary classification accuracy of 55% and 65% when tested with Hate Speech and Diplomacy. SAMsum is one of the datasets which covered a large area in the 2D plane shown in Figure 7.1; it lies near Hate Speech, GitHub, and Diplomacy.

Table 7.3 provides an analysis of the cosine similarities of the embeddings. It shows the top two most similar and the least similar datasets for each dataset. The results are consistent with Figure 7.1 and the binary classification task (Figure 7.3), showing that SwDA and Teams are two of the least similar datasets. SAMsum and Hate Speech demonstrate a high similarity with almost all the datasets other than SwDA and Teams. SAMsum and Hate Speech are also the datasets that exhibit the poorest binary classification accuracy (see Figure 7.3) and similar n-gram frequency



Unscaled Highest Peaks Data (Avg % similarity by Hamming distance)				
Dataset1	Dataset2	Peak value4	Peak value5	Accuracy
SAMSum(m)	Diplomacy(d)	2.608	0.958	0.65
SAMSum(m)	GitHub(g)	2.466	0.548	0.8
AMI (aim)	SAMSum(m)	2.363	0.936	0.85
SAMSum(m)	Hate Speech(h)	3.207	1.459	0.55

Unscaled Lowest Peaks Data (Avg % similarity by Hamming distance)				
Dataset1	Dataset2	Peak value4	Peak value5	Accuracy
GitHub(g)	SwDA(s)	0.411	0.057	0.85
Diplomacy(d)	SwDA(s)	0.411	0.116	0.95
GitHub(g)	Teams(t)	0.566	0.091	0.95
Teams(t)	SwDA(s)	0.393	0.114	100

Figure 7.3: The trend of binary classification accuracy (for the SVM RBF kernel) vs. average percentage similarity (normalized in the illustration) using the Hamming distance of length 4 and 5 subsequences. Hamming distance similarity predicts poor classification accuracy at the dataset discrimination task. This does not include the results for the Military dataset; its small test set gave 100% accuracy on all the datasets.

distributions (see Table 7.1) with one another. In general, social media datasets exhibit a high degree of similarity.

Table 7.3: The top two most similar and least similar datasets according to cosine similarity. The cosine similarity for some cases is negative because it is calculated between the embeddings generated through Doc2Vec, not using TF-IDF.

Dataset	**Most Similar**	**2nd Most Similar**	**Least Similar**
Army(a)	h(0.4528)	m(0.4494)	s(-0.0526)
AMI(ami)	h(0.4043)	m(0.2880)	s(0.0966)
Diplomacy(d)	h(0.4511)	m(0.4194)	t(-0.0126)
GitHub(g)	h(0.2753)	d(0.2534)	a(0.0285)
Hate(h)	m(0.5281)	a(0.4578)	t(0.1062)
SAMSum(m)	h(0.5352)	a(0.4606)	s(0.0409)
SwDA(s)	h(0.1092)	ami(0.1034)	t(-0.0912)
Teams(t)	ami(0.1464)	h(0.1033)	s(-0.0924)

Perturbation Analysis

Noise was introduced into the data by performing two perturbations: 1) random deletion and 2) tag swapping. We randomly swap 10% of the tags of each dialogue and generate nine sequences per dialogue. Similarly, we randomly delete 10% tags to generate nine sequences per sequence. This data augmentation strategy is used to create larger but noisier datasets of dialogue act sequences. We resample the datasets according to the size of the Military dataset and select 140 sequences from each for analysis.

Figure 7.4 shows the comparison of binary classification accuracy with or without perturbation. The results on the actual dataset vs. the perturbed one show large decreases in the classification accuracy of some of the datasets due to the noise. Altering dialogue act patterns causes the dataset to become similar to some of the other datasets. Figure 7.5 shows the distribution of synthetic dataset embeddings on a 2D plane. Compared to the embeddings of the original dataset, synthetic dataset embeddings of Diplomacy, Teams, and Army show a slight change in distribution and decreased accuracy with some of the datasets. The formal discussion (AMI and GitHub) perturbed

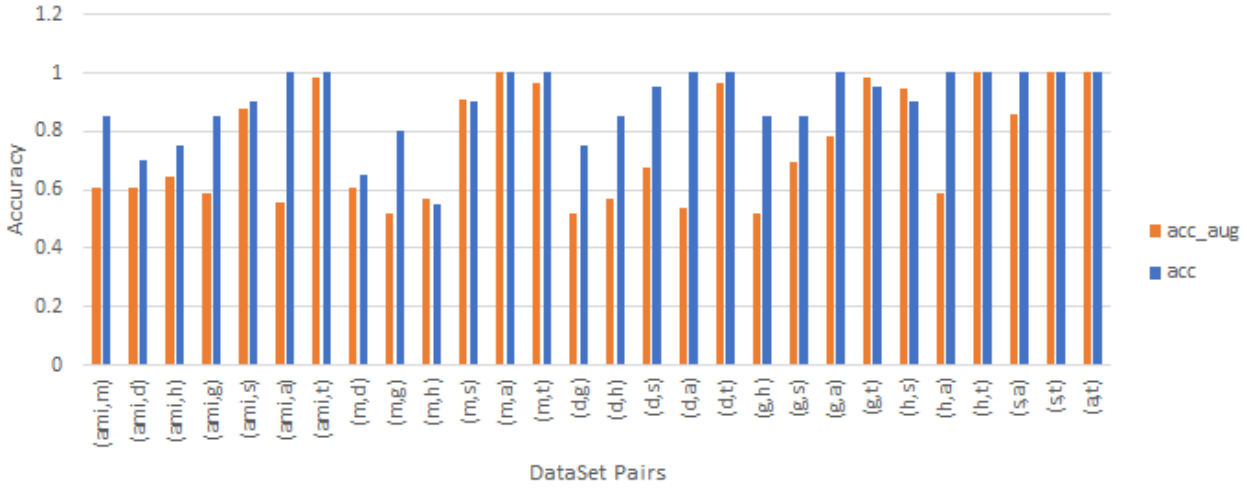


Figure 7.4: Comparison between the binary classification accuracy of synthetically perturbed data (acc_aug) and actual data (acc). ami: AMI, g: GitHub, d: Diplomacy, t:Teams, h: Hate Speech, s: SwDA, m: SAMsum, a: Military (Army).

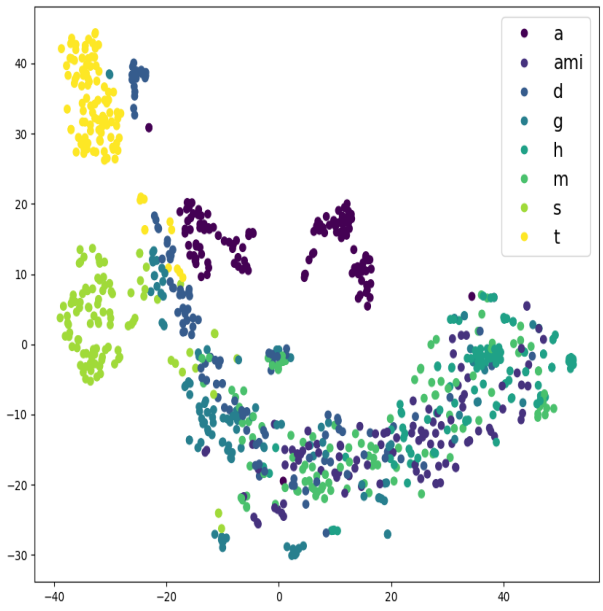


Figure 7.5: Projection of perturbed dataset embeddings in 2D space. ami: AMI, g: GitHub, d: Diplomacy, t:Teams, h: Hate Speech, s: SwDA, m: SAMsum, a: Military (Army)

datasets showed a greater decrease in classification accuracy than others.

Even in the presence of noisy data, the overall distribution of synthetic datasets embeddings, given by Figure 7.5, is still similar to the embeddings of original datasets (see Figure 7.1). The learned embedding is clearly robust to slight perturbations.

Discussion and Conclusion

This chapter presents a dialogue act similarity analysis across multiple communication domains by calculating n-gram frequency distribution, Hamming distance, and the Doc2Vec embedding between dialogue act sequences. It is clear that dialogue act sequences can differ greatly when collected from different communication settings, but even dialogues collected from the same domain can exhibit different communication patterns. The discourse is clearly dependent on the nature and purpose of the conversation. Simple data augmentation techniques like random swap and random deletion tend to alter the dialogue flow such that it becomes more similar to other dialogue categories.

Among all the domains used for the analysis, social media datasets exhibited the highest degree of similarity with one another. Models learned on non-goal oriented discussion do not show potential to generalize well to goal-oriented task specific discussions, and vice versa. One of the most widely used datasets, SwDA, does not exhibit discourse patterns similar to the other datasets used in our analysis. Formal discussions seemed to follow a communication pattern that overlaps with other datasets, and the models learned on these datasets showed a potential to generalize better.

The analysis indicates that the selection of appropriate source and target datasets is equally crucial as developing efficient techniques to achieve generalizability in dialogue and discourse. Based on our analysis, it is problematic to assume that machine learning models trained on one type

of discourse will generalize well to other settings, due to contextual differences. We believe our Hamming distance similarity measure can be used to anticipate potential degradation in the performance of learned models during domain adaptation and to select compatible source and target datasets.

CHAPTER 8: MULTI-FEATURE EMBEDDING: A STEP TOWARDS GENERALIZABLE CONFLICT PREDICTION MODEL (H6)

This chapter includes contents and figures from the paper titled ”Enayet, A., & Sukthankar, G. (2023, May). Improving the Generalizability of Collaborative Dialogue Analysis With Multi-Feature Embeddings. In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (pp. 3533-3547).”.

Conflict prediction in communication is integral to the design of virtual agents that support successful teamwork by providing timely assistance. The aim of my dissertation is to analyze discourse to predict collaboration success. Unfortunately, resource scarcity is a problem that teamwork researchers commonly face since it is hard to gather a large number of training examples. To alleviate this problem, this chapter introduces a multi-feature embedding (MFeEmb) that improves the generalizability of conflict prediction models trained on dialogue sequences. MFeEmb leverages textual, structural, and semantic information from the dialogues by incorporating lexical, dialogue acts, and sentiment features. The use of dialogue acts and sentiment features reduces performance loss from natural distribution shifts caused mainly by changes in vocabulary.

This chapter demonstrates the performance of MFeEmb on domain adaptation problems in which the model is trained on discourse from one task domain and applied to predict team performance in a different domain. The generalizability of MFeEmb is quantified using the similarity measure proposed by Bontonou et al. [13]. Our results show that MFeEmb serves as an excellent domain-agnostic representation for meta-pretraining a few-shot model on collaborative multiparty dialogues.

Introduction

For many natural language processing applications, the ability to learn features that generalize well across multiple datasets is a key desideratum [97]. This study introduces a new multi-feature embedding, MFeEmb, that increases the generalizability of models learned from collaborative multiparty dialogues. Dialogues are different from single-author documents in that, along with textual information, they contain communication patterns that may serve as indicators of social dynamics. Treating a dialogue as a mere text collection ignores valuable information. We advocate exploiting implicit features present in multiparty dialogues that are less vulnerable to distribution shifts resulting from task domain changes.

This dissertation demonstrates the usage of MFeEmb on a communication analysis task: conflict prediction. Teamwork research faces a challenge of resource scarcity since the human subjects datasets are quite small (less than 100 samples), due to the difficulty of recruiting teams and the time consuming nature of many group tasks. A variety of social phenomena have been investigated within team communication research including entrainment [89] and emergent leadership [65]. Frequency of communication is not in itself a good predictor of team performance, but a meta-analysis conducted by Marlow et al. [67] that drew upon data from 150 studies conducted on 9702 teams concluded that high quality communication is positively related to team performance in many task domains. Conversely, process conflict, "disagreement among group members about the content of the tasks being performed, including differences in viewpoints, ideas, and opinions" [43], is usually negatively correlated with taskwork success.

Our aim is to be able to learn a model to classify process conflict from multiparty dialogues that generalizes well across multiple tasks. We treat the task of conflict prediction as a binary classification task with high conflict and low conflict being the two classes; the ground truth used by the conflict prediction model is measured using a post-task team process conflict survey. We con-

sidered three collaborative problem-solving tasks: software engineering, search and rescue, and cooperative gameplay.

Our proposed embedding, MFeEmb, leverages textual, structural, and semantic information from the dialogues by incorporating vocabulary, dialogue acts, and sentiment features. Lexical embeddings such as word2Vec and BERT [22] show good performance across multiple NLP tasks on in-domain test sets but are less robust to domain shift. Our experimental analysis identified that dialogue acts and sentiment sequences are informative features that predict conflict reliably even at the earliest stage of team problem-solving [28]; however classifiers constructed using these features still experience lackluster transfer performance when applied to new datasets, particularly when detecting high conflict examples [29].

To address this transfer problem, we propose the usage of MFeEmb, specifically as a meta-pretraining representation to be used within a few-shot model. MFeEmb combines the strengths of both domain-invariant and domain-specific features. This dissertation compares the generalizability potential of the MFeEmb embedding vs. standard word embeddings using inter-class and intra-class based similarity measures, proposed by Bontonou et al. [13]. Then we evaluate the performance of MFeEmb in a domain adaptation scenario in which the model is trained on discourse from one task domain and used to predict conflict in a different domain. Our results show that:

1. MFeEmb demonstrates superior generalizability over other embeddings for collaborative multiparty dialogues.
2. MFeEmb is an excellent representation choice for the meta-training stage of few-shot learning.
3. The domain adaptation performance of MFeEmb can be easily enhanced by task specific synonym replacement.

Background

Previous studies on group interaction tasks such as conflict prediction [89], disruptive talk detection [84], group satisfaction [55], and task performance prediction [52, 78] have focused on simply improving performance on in-domain datasets. Very little attention has been paid to the problem of creating generalizable models for multiparty dialogue that can be used when training data is scarce. The intelligent tutoring system community has empirically assessed the generalizability of common natural language representations, such as BERT and Linguistic Inquiry Word Count (LIWC), across collaborative problem solving tasks [88], but without investigating methods to improve generalizability.

In domain adaptation, the goal is to train a model on data from a source domain that performs well on a test dataset drawn from a different target distribution. Common NLP tasks (e.g., part-of-speech (POS) tagging and named entity recognition (NER)) have been tackled using techniques including instance weighting [45] or explicitly identifying feature correspondences between the domains [12]. An alternate approach is to learn a single representation that generalizes well across multiple domains. This can be done using few-shot learning [124], one of the most widely used approaches to dealing with resource scarcity. The traditional framework comprises meta-training and meta-testing phases, where the aim of meta-training is to learn universal representations from multiple domains.

Rather than seeking to learn the new representation entirely from data, our research exploits similarities in dialogue act sequences and sentiment patterns commonly observed during successful collaborative problem-solving.

Representation choice has been shown to place an upper bound on target domain performance [8]. Few shot frameworks such as Meta-pretraining then Meta Learning (MTM) [21] have assumed that

word embeddings like BERT that are trained on large datasets are the best choice for task agnostic pre-training. Bontonou et al. [13] introduced a method to quantify the generalizability of a few-shot classifier under supervised, unsupervised and semi-supervised settings. We use their inter-class and intra-class based generalizability measure to evaluate MFeEmb vs. simple word-based embeddings under supervised classification scenarios. Our research demonstrates that MFeEmb is superior to word embeddings as a meta- pretraining representation.

Methodology

This section describes our approach to learning a generalizable embedding from multi-party dialogues for conflict prediction. We discuss our datasets, introduce our embedding, and show how our technique can be used in combination with data augmentation and few shot learning.

Datasets

Datasets collected from different collaborative problem-solving task domains were used in our study of generalizability:

1. **Teams corpus** [61]: This dataset consists of dialogues from 62 teams playing a cooperative board game in groups of three or four. Each team plays the game twice together. The Teams corpus was originally created to study entrainment, a linguistic phenomena in which teammates adopt similar speech patterns [89]. The Game1 dataset of Teams corpus contains 62 dialogues, 32 low conflicts, and 30 high conflict dialogues. The Game2 dataset of Teams corpus contains 62 dialogues, 33 low conflicts, and 29 high conflict dialogues.
2. **ASIST dataset** [40]: This dataset consists of 67 teams of three people participating in a simulated search and rescue task within the Minecraft game environment. Participants com-

pleted two different missions that involved searching a map and triaging victims. The dataset was collected by the ASIST project to stimulate the development of proactive assistant agents for helping human teams. The dataset contains 113 dialogues, 58 low conflicts, and 55 high conflict dialogues.

3. **GitHub social coding dataset** [27]: This dataset was mined directly from the GitHub social coding platform. It consists of data from issue comments of teams developing open source software over a period of months. Teams vary in size, and comments were harvested for 50 reported issues. The dataset contains 50 dialogues, 29 low conflicts, and 21 high conflict dialogues.

Both the Teams and ASIST datasets contain post-task process conflict survey data for all teams, which we divide into high and low conflict groups using their z-scores. For GitHub, process conflict was scored according to issue resolution using the following heuristics to determine if conflicts occurred:

1. Unsuccessful resolution of the issue.
2. Unanswered questions in the discussion.
3. Lack of understanding about the issue from one or more members.
4. Lack of understanding or disagreement between the team members.
5. Disagreement between the members about the proposed solution.

Multi-Feature Embedding (MFeEmb)

This dissertation introduces the MFeEmb embedding which is designed to capture the dialogues' structural, semantic, and textual information for collaborative task success prediction. To represent the structural information, we incorporate information from dialogue acts (DAs) of the utterances. For semantics, the sentiment polarities of the utterances are used, although DAs capture both se-

mantic and structural information. Textual information is extracted from the vocabulary of the dialogues.

For the word embedding, we use both the Distributed Bag of Words and Dynamic Memory models of Doc2Vec [57] to learn embeddings. Although there is only 28% vocabulary overlap between the ASIST and Teams datasets and 35% overlap between the GitHub and Teams datasets, word embeddings can help preserve high performance on the training dataset, while including structural and semantic features makes the embedding more robust to domain shifts.

For the dialogue act (DA) embedding, we first map the sequence of utterances to a sequence of DAs using our USE-DAC (Universal Sentence Encoder Dialogue Act Classifier, described in Chapter 4). The SwDA-DAMSL tagset was used to categorize dialogue acts. The TextBlob python module was used to assign sentiment polarities ranging from -1 to 1 to each of the utterances.

To generate the embeddings, we use the Dynamic Memory model of Doc2Vec due to the small vocabulary size of the sequences, which is limited by the number of DA tags and sentiment gradations. The Dynamic Memory model leverages context when generating embeddings, thus preserving information contained in these communication patterns. In contrast, the Distributed Bag of Words model does not consider context when generating embeddings. For the few-shot results, we also report results with pre-trained Word2Vec embeddings. First, we separately learn three embeddings from the sequence of DAs, sentiments, and utterances (text); the final MFeEmb embedding is created either by concatenating the three embeddings or by using LSTMs to learn a concatenation ensemble model (see Figure 8.1).

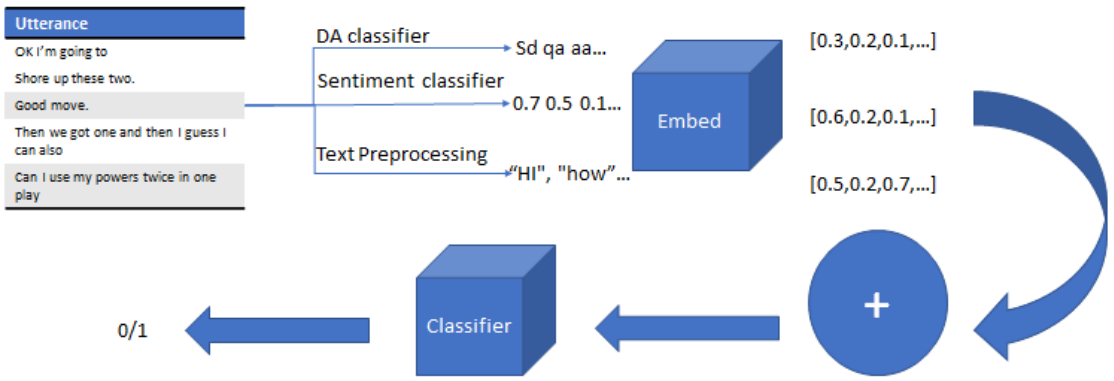


Figure 8.1: Utterances are classified using the dialogue act classifier to produce a sequence of DAs and the sentiment classifier to produce a time series of sentiment polarities. Along with the text data, these sequences are used to create MFeEmb using the Dynamic Memory model of Doc2Vec. The few shot learning and data augmentation options are not shown in the figure.

Corpus-Based Feature Analysis

To understand the ramifications of our feature selections, we performed frequency distribution analyses across the high conflict and low conflict classes of the Teams Dataset. This analysis shows that the high conflict class has a high frequency of negative sentiment polarities compared to the low conflict class and a comparable frequency of positive sentiment polarities compared to the low conflict class (Figure 8.2).

In the dialogue act distribution, Statement-non-opinion (sd) is the most frequent tag in both classes. The low conflict class has a high frequency of positive communication indicators like Appreciation (ba), Conventional-closing (fc), and Thanking (ft) compared to the high conflict class. The high conflict class contains a high frequency of bad communication indicators like Uninterpretable (%), Hedge (h), Signal-non-understanding (br), and Apology (fa). Interestingly, high conflict classes have a high frequency of all categories of questions compared to low conflict classes (see dialogue

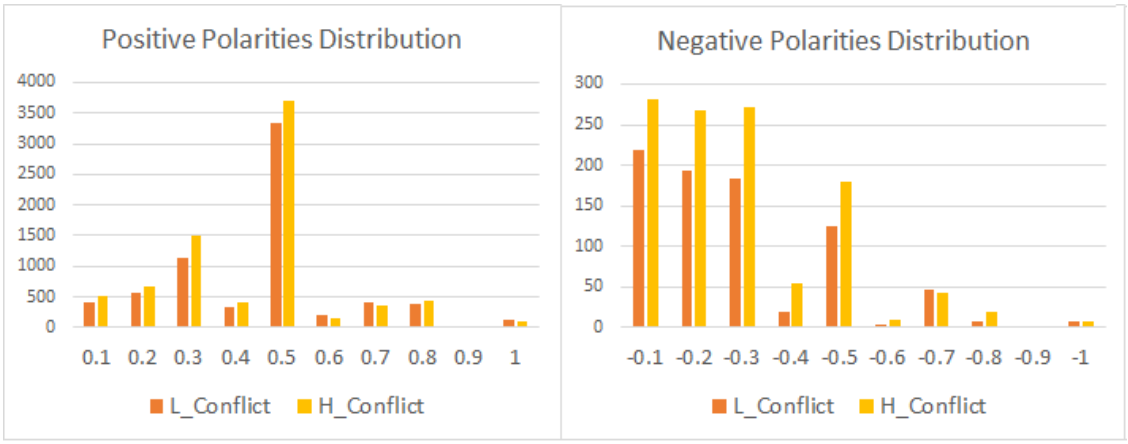


Figure 8.2: Sentiment polarity distribution of the high conflict vs. low conflict classes in the Teams dataset

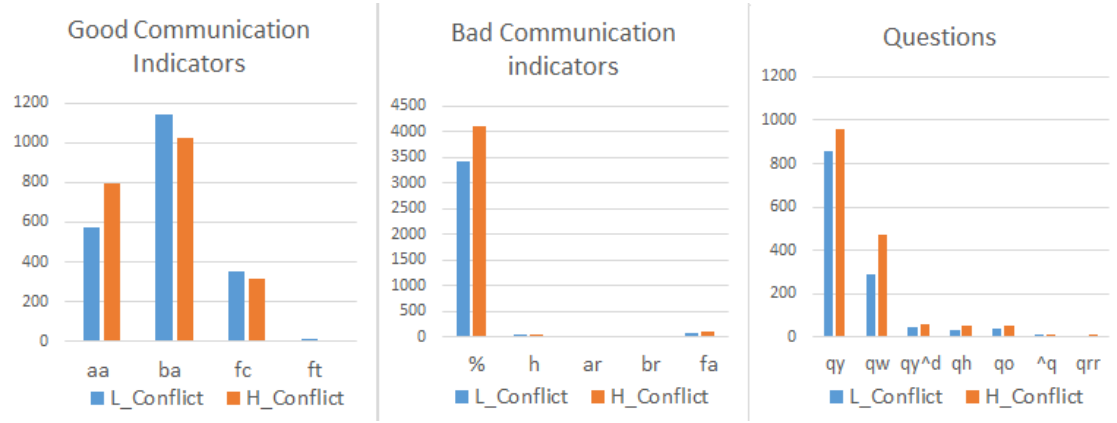


Figure 8.3: Dialogue Acts frequency distribution of the high conflict vs. low conflict classes in the Teams dataset

act distributions and n-grams in Figure 8.3.

Looking at the vocabulary distribution, the high conflict class contains more profanity words than the low conflict class, and there is no overlap between the profanity word lists of both classes. The most frequent words in the high conflict dialogues that are in profanity list are: 'hell', 'kill', 'suck', 'sucking', 'shit', 'strip', 'stroke', 'rectum', 'xxx', 'dick', 'screwed', 'retard', 'ovary', 'piss',

'lube', 'junkie'. The most frequent words in the low conflict dialogues that are in the profanity list are: 'booty', 'pot', 'carpet', 'rum', 'breasts', 'pedophile', 'urine', 'thug', 'screw', 'jerk', 'weed', 'screwing', 'shower', 'stupid'.Our analysis reveals that there is value in all three types of features (dialogue acts, sentiment polarity, and vocabulary) but that conflict prediction remains a challenging classification problem.

Synthetic Datasets

To further improve generalization, we augment our training data with synthetic datasets generated using synonym replacement, as proposed by Wei and Zou [127]. Our data augmentation strategies are described below:

1. **SynReplace**: We augment Teams Game1 and Game2 by replacing the words with synonyms drawn from WordNet.
2. **ASISTReplace**: We augment Teams Game1 and Game2 by replacing the words with only the synonyms present in the ASIST dataset. First, we extract the vocabulary of the ASIST dataset. During the replacement operation, we search for synonyms in WordNet and only replace them with the synonyms present in the ASIST dataset's vocabulary.
3. **GitReplace**: Similar to ASISTReplace, we generate our third dataset by replacing the words with only the synonyms present in the GitHub dataset.

Four synthetic dialogues are generated for each dialogue of the Teams dataset after applying random replacement on 10% of the words. Our intuition is that collaborative problem-solving domains such as software engineering may contain a lot of task specific jargon, and even simple synonym replacement techniques greatly facilitates generalization.

In our experiments, the basic synonym replacement did not significantly change the intent and

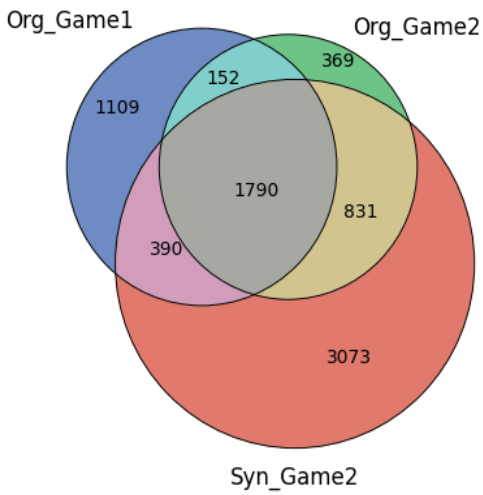


Figure 8.4: Vocabulary overlap between original Game1, original Game2, and the Game2 adversarially generated dataset

sentiment of the utterances. To show the robustness of dialogue acts and sentiment sequences towards data augmentation, we utilize TextAttack [76], a python package for adversarial attack and data augmentation, to generate a Teams Game2 synthetic dataset. Word Swap by BERT-Masked LM transformation was employed to generate synthetic examples from the Teams Game2 dataset. One synthetic example is generated per dialogue of the Game2 dataset. The synthetic dataset contains $\approx 50\%$ more unique words than original Game2 dataset (Figure 8.4). Hamming distance was used to calculate the difference between the sequences of the Game2 original and Game2 synthetic datasets. On average, the adversarial synthetic dataset only resulted in a 11% change in DA sequences and 14% change in sentiment sequences.

Experimental Setup

The Teams corpus contains 124 team dialogues from 62 different teams, playing two different collaborative board games. We use the Teams Game1 dataset with 62 total samples, divided into

32 low and 30 high conflict samples, as our training dataset. The small training dataset ensures that the experiments reflect the generalization performance under the resource scarcity scenario. Our test datasets for evaluating domain adaptation are Teams Game2, GitHub, and ASIST. Obviously the domain shift is the smallest between the Teams Game 1 and 2 datasets. We use the GitHub and ASIST datasets to check the transferability of MFeEmb under domain shift. The model was not fine-tuned before evaluating the performance on GitHub and ASIST. We selected these three datasets for testing because they reflect different levels of transfer complexity, starting from the least complex transfer problem, i.e., Teams Game2 dataset, to the most complex ASIST dataset, based on the level of natural distribution shift (Figure 8.5). We evaluate our proposed MFeEmb

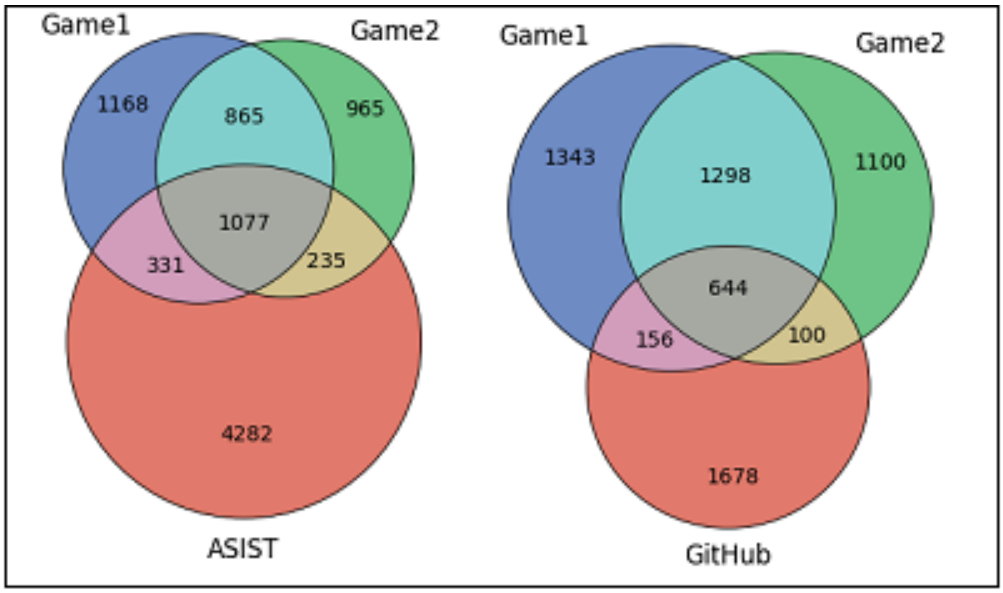


Figure 8.5: Vocabulary overlap between datasets.

under the following three experimental setups:

1. SVM and logistic regression classifiers to distinguish high conflict and low conflict classes.
2. LSTM concatenation ensemble.
3. Few-shot learning approach.

We benchmark MFeEmb against prior work on conflict prediction, other embedding choices, and FsText, a few-shot model proposed by Bailey and Chopra [7]. Experiments were performed using a 300-dimensional version of MFeEmb where the length of all the three embeddings is the same, i.e., 100. We report the mean and standard devication of F1-Scores after 15 runs. For the SVM and logistic regression classification experiments, we only report the results (mean F1-Scores) of the best performing classifier. '*' denotes that logistic regression was the top performer, and '+' denotes cases where the SVM was the best.

SVM and Logistic Regression

After Doc2Vec is used to generate the three embeddings for each sample, the embeddings are concatenated to create MFeEmb. We use both SVM and logistic regression to classify the instances and report the result of the one that shows the highest accuracy. For DAs and sentiment sequences, we always use the Dynamic Memory model (DM) of Doc2Vec.

Few-Shot Learning (FsText)

For few-shot learning, we use the method proposed by Bailey and Chopra [7] and available in the FsText Python module. The training document for the meta-training stage of few-shot learning is represented using a pre-trained word embedding (Word2Vec). In the case of more than one training sample per class, the proposed method works by averaging each class's vectors to calculate the most effective class representative. Cosine similarity is used to measure the distance between the test sample and each class representative, and the test sample is assigned the label of the class with the highest similarity. We compare the generalizability of FsText (Original) with MFeEmb-based FsText, by replacing Word2Vec embedding with MFeEmb during the meta-training stage.

Concatenation Ensemble

Due to the small size of the training set, we apply the synonym replacement technique proposed by Wei and Zou [127] to augment the training data. One hot encoding is used to encode DA, sentiment polarities, and vocabulary to train the model. We train three different Bidirectional LSTM models, one on each of DAs, sentiments, and word-based documents, and merge them to create our MFeEmb based ensemble. Our Bidirectional LSTM models for each feature have an embedding layer, an LSTM layer, one dropout layer, and one deep layer.

Baseline Models

We compare our proposed MFeEmb's results with several baseline models that use the same binary classification setup for conflict prediction. First, we show that MFeEmb performs competitively against prior work on conflict prediction [28] using their proposed dialogue act only and sentiment only embeddings. Note that our results are not directly comparable to what was reported in their paper because we use a reduced training set thus we have reimplemented their embeddings.

We also compare MFeEmb to the commonly used BERT based embedding. We use the bert_en_uncased_L12_H768_A12 model available at TensorFlow Hub[1] to develop our baseline classifier. The model contains one dense layer, one dropout layer, a sigmoid activation function, Adam optimizer. Due to the small size of the Game1 dataset we train the model on the synonym replaced Game1 dataset.

These independent baselines are compared against three implementation options for MFeEmb: 1) MFeEmb with simple binary classifier (SVM or logistic regression), 2) MFeEmb concatenation ensemble learned with LSTMs trained on the synonym replaced augmented dataset, 3) a variation

[1] https://tfhub.dev/tensorflow/bert_en_uncased_L-12_H-768_A-12/4

of few-shot learning method (FsText) [7] in which the Word2Vec embedding is replaced with MFeEmb during the meta-training stage. For training and testing, we concatenate all the utterances of the dialogue into one single document and assign it to one of the classes depending on the conflict score of the team.

Results

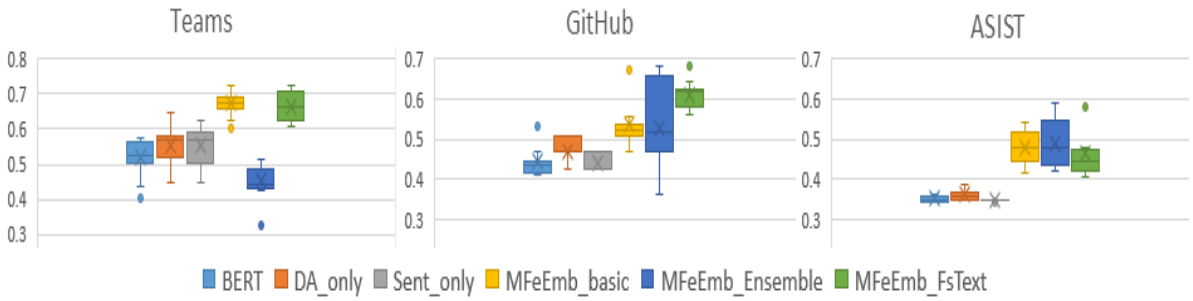


Figure 8.6: Performance of MFeEmb vs. other embedding choices from prior work.

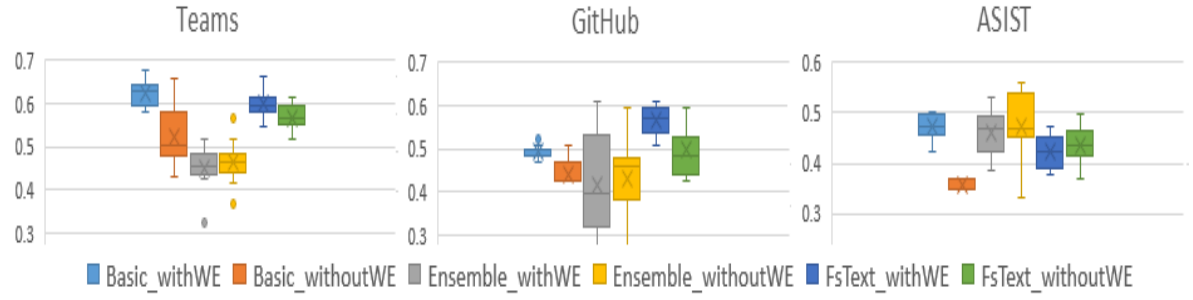


Figure 8.7: Performance of MFeEmb with and without word embedding (WE).

This section presents results on the generalizability of MFeEmb under different experimental setups.

Similarity Based Evaluation

First we quantify the potential generalization of the representation using the similarity measure proposed by Bontonou et al. [13]. The similarity measure is given by:

$$intra(c) = \frac{1}{k(k-1)} \sum_{iy_i=c} \sum_{j \neq iy_j=c} \cos\left(f_i, f_j\right) \tag{8.1}$$

$$inter(c, \tilde{c}) = \frac{1}{k^2} \sum_{iy_i=c} \sum_{j \neq iy_j=\tilde{c}} \cos\left(f_i, f_j\right) \tag{8.2}$$

$$similarity = \frac{1}{N} \sum_{c=1}^{N} (intra(c) - \max_{c \neq \tilde{c}}(inter(c, \tilde{c}))) \tag{8.3}$$

where c is class, N is the number of classes, k is number of examples, f is the embedding, $intra(c)$ is cosine similarity within a class, and $inter(c, \tilde{c})$ is cosine similarity through classes c and $\tilde{c}$. The final similarity score reflects the comparison of the $intra(c)$ and $inter(c, \tilde{c})$. Intuitively it can be seen that the score measures how the representation affects the data clustering within and between classes.

We compare our proposed MFeEmb vs. a standard word embedding learned using the bag of word model of Doc2Vec. Table 8.1 gives the result of the similarity-based analysis, juxtaposed with the classification results. MFeEmb has a better similarity score and high classification performance, compared to word-based embeddings indicating the high generalizability potential of MFeEmb. Figure 8.8 shows the visualization of the two embeddings projected on a 2D plane.

MFeEmb Performance Summary

Figure 8.6 provides the overall comparison of MFeEmb vs. the benchmark embeddings. In the case where minimal domain adaptation was required (testing classifiers on Teams2 that were trained on

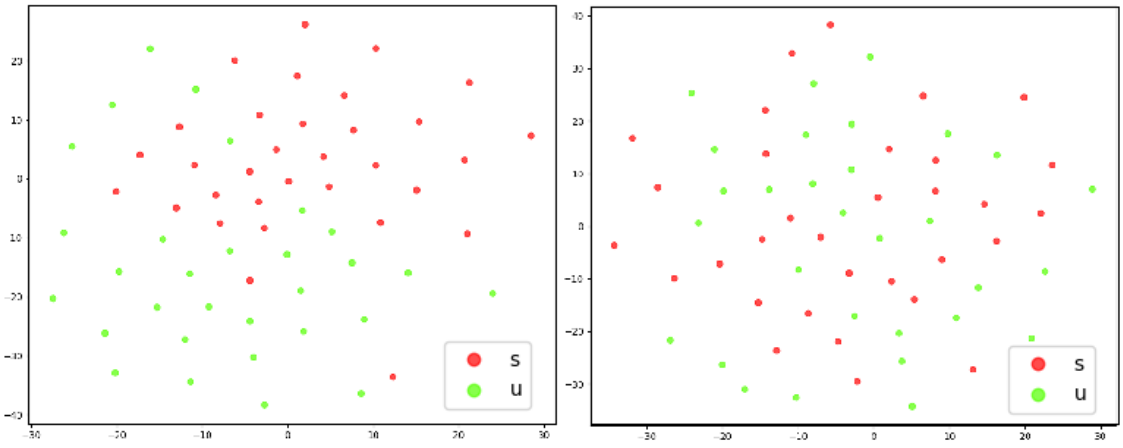


Figure 8.8: Comparison of the MFeEmb (left) and word embedding (right) distribution on the 2D plane. Multi-feature embedding showed better clustering, with most instances of one of the classes occupying the lower left and the other occupying the upper right. On the other hand, word embeddings are very intermixed. s: low conflict (successful dialogue), u: high conflict (unsuccessful dialogue).

Teams1), the simple version of MFeEmb using a SVM classifier is the top performer and outperforms the embeddings used in other prior work on conflict prediction [28]. Our most consistent model, MFeEmb with FsText, showed a significantly high F1-Score on high conflict class compared to baseline models (see Table 8.2). Note that detecting the high conflict examples is more valuable for practical implementations.

For the more complex domain adaptation scenarios (GitHub and ASIST), the best performance was achieved using MFeEmb as a replacement for the Word2Vec embedding during the meta-training phase of FsText on GitHub, and the concatenation ensemble showed significantly better performance on the ASIST dataset. The vanilla MFeEmb generally performed comparably to the concatenation ensemble using LSTMs on out of domain datasets. The latter showed a high standard deviation compared to the former.

To analyze the importance of incorporating word embedding in MFeEmb, we compare the per-

formance of all the experimental setups with and without word embedding (WE). For SVM & Logistic regression (Basic) and FsText, we train the model on the Teams Game1 dataset, and for concatenation ensemble, we train on the synonym replaced dataset. One of our main objectives in incorporating the word embedding in MFeEmb is to maintain the performance on the in-domain dataset, and results show that MFeEmb performed better with word embedding on the in-domain dataset. For most transfer case setups, MFeEmb with word embedding either gave better or comparable mean F1 scores (Figure 8.7). The following sections present a more in-depth evaluation of each experimental setup.

Table 8.1: Similarity-based generalizability analysis.

Word_Emb		**MFeEmb**	
Teams Game2			
similarity	F1_score	similarity	F1_score
-0.067	0.470*	-0.016	**0.628+**
GitHub			
similarity	F1_score	similarity	F1_score
-0.067	0.463*	-0.017	**0.501+**
ASIST			
similarity	F1_score	similarity	F1_score
-0.067	0.348*	-0.016	**0.458+**

SVM and Logistic Regression

Table 8.3 gives the results for the SVM and logistic regression classifiers. This paper presents a thorough evaluation of the performance of different embedding choices (DM, DBOW).We also evaluate the performance of different data augmentation methods (**SynReplace**, **ASISTReplace**, and **GitReplace**).

Our proposed MFeEmb trained using Doc2Vec and classified using either SVM or logistic regres-

sion performed better than the word-embedding baseline. Leveraging synthetic datasets yielded significant performance improvements. In our most challenging resource-scarce scenario, where we trained the model only on the Teams Game1 dataset, incorporating word embedding showed better performance on the Teams Game2 and GitHub datasets, while the model performed better on the ASIST dataset without word embedding (see Figure 8.7).

Table 8.2: Summary of high conflict class F1_scores

High Conflict Class Prediction Summary		
Method	**GitHub**	**ASIST**
BERT_SynReplace	0.431	0.347
DA_only_Team1	0.320*	0.311*
Senti_only_Team1	0.207*	0.300*
MFeEmb_FsText_Team1	**0.564**	**0.478**

Concatenation Ensemble Model

Table 8.3 gives the results for the LSTM-based concatenation ensemble model. The model showed a better mean F1-score than the text-based LSTM model. We also trained the LSTM using synthetic datasets generated using GitHub and ASIST vocabularies, which showed better performance, specifically with the GitHub vocabulary dataset. The model performed significantly better on the ASIST dataset compared to the other experimental setups.

Few-Shot Model (FsText)

The FsText baseline showed the best performance on Game2, but the performance degraded considerably on the transfer task (GitHub and ASIST). FsText with the proposed MFeEmb exhibited significantly better performance on the GitHub and ASIST datasets, specifically with ASIST vo-

cabulary's synthetic dataset. FsText with the proposed MFeEmb embedding also gave a comparable performance on the Teams Game2 dataset. This demonstrates that MFeEmb is an excellent representation for meta-pretraining a few shot model on collaborative multiparty dialogues even when learned from a small dataset (see Table 8.3).

Using a synthetic dataset showed a performance improvement in all three experimental setups. Generation of the synthetic dataset using the vocabulary of other collaborative tasks showed comparatively better performance on the transfer task. Even in the in-domain experiments, the Game1 Synthetic dataset, generated using collaborative task vocabulary, showed the best and comparable performance on Game2 in all the experimental setups.

Results on Adversarially Generated Dataset

This section presents results on the adversarially generated dataset (Synthetic Game 2) created using TextAttack[2]. Word Swap by BERT-Masked LM transformation was employed to generate synthetic examples from the Teams Game2 dataset. One synthetic example is generated per dialogue of the Game2 dataset. The length of the synthetic Game2 dataset vocabulary is 6084, and the length of the original Game1 dataset vocabulary is 3441. The number of words in the synthetic dataset that are not in the original Game1 is 3904.

Figure 8.4 shows a high overlap between original Game1 and original Game2 compared to synthetic Game2 and original Game1, but this does not affect the performance of MFeEmb (Basic), and MFeEmb (Basic) gave a better performance on the synthetic dataset. On the other hand, the performance of the BERT baseline decreased on the synthetic Game2 test set, with a high standard deviation in mean F1 scores (see Table 8.4).

[2]`https://github.com/QData/TextAttack`

Table 8.3: Detailed performance evaluation of MFeEmb.

SVM & Logistic Regression Results			
Method	**Teams Game2** F1_score (std)	**GitHub** F1_score (std)	**ASIST** F1_score (std)
Baseline Doc2Vec_dbow	0.465 (0.070)*	0.489 (0.080)*	0.425 (0.091)*
MFeEmb_Team1_dbow	0.533 (0.068)*	0.437 (0.025)*	0.347 (0.002)*
MFeEmb_Team1_dm	0.625 (0.0295)+	0.495 (0.012)+	0.473 (0.023)+
MFeEmb_SynReplace	0.558 (0.035)+	0.296 (0.025)*	0.318 (0.00)*+
MFeEmb_GitReplace	**0.676 (0.033)+**	0.409 (0.039)*	0.411 (0.041)*
MFeEmb_ASISTReplace	0.675 (0.041)+	**0.537 (0.060)***	**0.480 (0.042)***
Concatenation Ensemble Results			
Baseline_SynReplace	0.435 (0.048)	0.414 (0.104)	0.397 (0.081)
MFeEmb_SynReplace	0.453 (0.044)	0.429 (0.122)	0.459 (0.044)
MFeEmb_GitReplace	**0.464 (0.044)**	0.468 (0.098)	**0.491 (0.054)**
MFeEmb_ASISTReplace	0.408 (0.075)	**0.516 (0.100)**	0.455 (0.059)
Few Shot Learning Results			
FsText Baseline	**0.689 (0.0)**	0.330 (0.0)	0.338 (0.0)
MFeEmb_Team1_doc2Vec	0.60 (0.028)	0.583 (0.045)	0.451 (0.025)
MFeEmb_Team1_word2Vec	0.597 (0.041)	0.507 (0.063)	0.437 (0.027)
MFeEmb_SynReplace	0.544 (0.021)	0.568 (0.031)	0.435 (0.037)
MFeEmb_GitReplace	0.684 (0.033)	0.567 (0.041)	0.388 (0.266)
MFeEmb_ASISTReplace	0.664 (0.042)	**0.608 (0.034)**	**0.462 (0.053)**

Table 8.4: MFeEmb results on the Game2 synthetic dataset generated using TextAttack.

Game2 Synthetic Dataset Results				
Train	**Teams Game1**	**SynReplace** F1_score (std)	**GitReplace** F1_score (std)	**ASISTReplace** F1_score (std)
MFeEmb	0.654 (0.033)+	0.443 (0.046)*	**0.617 (0.035)+**	**0.624 (0.055)+**
BERT	-	**0.490 (0.061)**	0.422 (0.037)	0.495 (0.044)

Conclusion

This chapter introduces our proposed multi-feature embedding (MFeEmb), a combination of textual (words), structural (DAs), and semantic (sentiment, DAs) embeddings to reduce the performance loss due to natural distribution shift. Experiments show that the multi-feature embedding performs significantly better than sentence (BERT), dialogue act-only, sentiment-only, and word embeddings. Our results demonstrate that MFeEmb is a superior representation for meta-pretraining a few-shot model that works well across different collaborative problem-solving domains.

Our proposed data augmentation strategy successfully resolved the domain shift problem caused by task-specific vocabulary without perturbing the dialogue act and sentiment features. Experiments with synthetic datasets show that synonym replacement with vocabulary drawn from a collaborative task outperforms generic synonym replacement with WordNet. It improves both the transfer accuracy and the test accuracy on the in-domain test set. Note that we did not fine-tune the models on the target datasets, i.e., GitHub and ASIST, and strictly report the model learned on the Teams dataset. Only the vocabulary of these datasets was used to boost the performance; explicit fine-tuning of the machine learning models could further improve the results.

CHAPTER 9: CONCLUSION

Collaborative problem-solving is integral to the successful completion of almost all tasks. With the advent of technology and online collaborative platforms like Microsoft Teams, GitHub, and Zoom, the mode of collaboration is now hybrid as opposed to entirely in-person. Virtual collaboration has its own challenges, sometimes leading to a conflict between the team members about the task being performed. Virtual agents can help in this regard by providing timely assistance, but for that, there is a need for proactive conflict detection in communication. First, this dissertation examines the utility of three different embeddings that we generate from 1) DAs, 2) sentiment polarities, and 3) entrainment on the task of proactive conflict detection. The experimental analysis shows that the DA embedding is more predictive of conflict during the early stages of the dialogues, followed by sentiment polarities which also show significant improvement over entrainment based embedding proposed by Rahimi and Litman [89].

Our research aims at utilizing limited resources and improving the generalizability, and in this regard, the first challenge we faced was developing a DA classification model with good transfer capabilities. We started our research by identifying an embedding model that could perform well under natural distribution shifts for the task of DA classification. We compared the performance of BERT, USE, Glove, and Probabilistic Embedding models and identified that USE with three dense layers gives the best transfer performance on the out-of-domain dataset. One of the contributions of this research is our GitHub dataset that we extracted from GitHub issue comments to test the transferability of the models.

Resource scarcity is a problem that most teamwork researcher face. The lack of resources to train an extensive machine learning model hinders the development of collaborative dialogue analysis models. Our second contribution is developing a method to improve the generalizability of the

conflict prediction model under resource-scarce scenarios. This dissertation introduces a multi-feature embedding (MFeEmb) to improve the generalizability of multi-party dialogue models under resource scarcity. The MfeEmb combines the strength of domain invariant and domain-specific features to improve generalizability.

For domain invariant features, we selected DAs and sentiment polarities since the vocabulary of these is limited to the number of tags used to train the classifier. We tested the generalizability of DA and sentiment on the out-of-domain datasets, and the results confirmed the utility of both features. Other features that we considered are speaker switches and entertainment. We didn't choose speaker switches because the number of speakers is not fixed, varies from team to team, and is not domain-invariant. On the other hand, entrainment completely failed the transferability test. For domain-specific features, we select the textual feature of the dialogue, i.e., the vocabulary of the dialogue.

We compared the performance of MFeEmb with DA-only, sentiment-only, BERT, and Few-shot models. MFeEmb significantly performed better than the baselines. For Few-shot learning, we compared the performance of the MFeEmb against the universal embedding and identified that MFeEmb is an excellent alternative to universal embedding that requires a large amount of data to train. MFeEmb performed significantly better than the universal embedding on the transfer task.

Third, to further improve the performance, we propose a data augmentation strategy, i.e., the generation of synthetic datasets using the vocabulary of the collaborative dialogues. This dissertation comprehensively compares the performance of conflict prediction models trained on the synthetic dataset generated using synonyms from WordNet with the synthetic dataset generated using synonyms from collaborative dialogue vocabulary. Results show that our proposed method to generate a synthetic dataset from vocabulary of collaborative dialogues significantly improves performance.

Fourth, this dissertation also proposes a method to measure the similarity between dialogue act

sequences of different dialogue domains to identify the possible degradation in the performance of the dialogue model due to natural distribution shifts.

This dissertation only reports results on the generalizability of MFeEmb on conflict prediction tasks; MFeEmb may not perform as well on other communication analysis tasks. However, we believe that modifying the features used in the embedding can address this problem. In future work, we are interested in applying our embedding to new team communication analysis tasks such as identifying emergent leadership.

LIST OF REFERENCES

[1] John Aberdeen and Lisa Ferro. Dialogue patterns and misunderstandings. In *ISCA Tutorial and Research Workshop on Error Handling in Spoken Dialogue Systems*, 2003.

[2] Samarth Agrawal, Aditya Joshi, Joe Cheri Ross, Pushpak Bhattacharyya, and Harshawardhan M Wabgaonkar. Are word embedding and dialogue act class-based features useful for coreference resolution in dialogue. In *Proceedings of PACLING*, 2017.

[3] Ali Ahmadvand, Jason Ingyu Choi, and Eugene 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*, pages 1273–1276, 2019.

[4] James Allen and Mark Core. Draft of DAMSL: Dialog act markup in several layers, 1997.

[5] N. Ambady and R. Rosenthal. Thin slices of expressive behavior as predictors of interpersonal consequences: a meta-analysis. *Psycholology Bulletin 111*, 2:256—274, 1992.

[6] N. Ambady and R. Rosenthal. Half a minute: predicting teacher evaluations from thin slices of non-verbal behavior and physical attractiveness. *J. Pers. Soc. Psychol.*, 64(3):431—441, 1993.

[7] Katherine Bailey and Sunny Chopra. Few-shot text classification with pre-trained word embeddings and a human in the loop. *arXiv preprint arXiv:1804.02063*, 2018.

[8] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006.

[9] Štefan Beňuš, Marian Trnka, Eduard Kuric, Lukáš Marták, Agustín Gravano, Julia Hirschberg, and Rivka Levitan. Prosodic entrainment and trust in human-computer interaction. In *Proceedings of the 9th International Conference on Speech Prosody*, pages 220–224. International Speech Communication Association Baixas, France, 2018.

[10] Timothy Bickmore and Daniel Schulman. Empirical validation of an accommodation theory-based model of user-agent relationship. In *International Conference on Intelligent Virtual Agents*, pages 390–403. Springer, 2012.

[11] Steven Bird, Branimir Boguraev, Martin Kay, David McDonald, Don Hindle, and Yorick Wilks. *Survey of the state of the art in human language technology*, volume 12. Cambridge University Press, 1997.

[12] John Blitzer, Ryan McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 120–128, Sydney, Australia, July 2006. Association for Computational Linguistics.

[13] Myriam Bontonou, Louis Béthune, and Vincent Gripon. Predicting the generalization ability of a few-shot classifier. *Information*, 12(1):29, 2021.

[14] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*, 2018.

[15] Lin Chen and Barbara Di Eugenio. Multimodality and dialogue act classification in the robohelper project. In *Proceedings of the SIGDIAL 2013 Conference*, pages 183–192, 2013.

[16] Xiao Chen, Gabriel Campero Durand, Roman Zoun, David Broneske, Yang Li, and Gunter

Saake. The best of both worlds: combining hand-tuned and word-embedding-based similarity measures for entity resolution. *BTW 2019*, 2019.

[17] Zheqian Chen, Rongqin Yang, Zhou Zhao, Deng Cai, and Xiaofei He. Dialogue act recognition via crf-attentive structured network. In *International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 225–234, 2018.

[18] Cristian Danescu-Niculescu-Mizil, Michael Gamon, and Susan Dumais. Mark my words!: linguistic style accommodation in social media. In *Proceedings of the International Conference on World Wide Web*, pages 745—754. 2011.

[19] Cristian Danescu-Niculescu-Mizil, Lillian Lee, Bo Pang, and Jon Kleinberg. Echoes of power: Language effects and power differences in social interaction. In *Proceedings of the International Conference on World Wide Web*, pages 699–708, 2012.

[20] Rajshekhar Das, Yu-Xiong Wang, and José MF Moura. On the importance of distractors for few-shot classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9030–9040, 2021.

[21] Shumin Deng, Ningyu Zhang, Zhanlin Sun, Jiaoyan Chen, and Huajun Chen. When low resource nlp meets unsupervised language model: Meta-pretraining then meta-learning for few-shot text classification. *arXiv preprint arXivi1908.08788*, 2019.

[22] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[23] Junhao Dong, Yuan Wang, Jian-Huang Lai, and Xiaohua Xie. Improving adversarially robust few-shot image classification with generalizable representations. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9025–9034, 2022.

[24] Nathan Duran and Steve Battle. Probabilistic word association for dialogue act classification with recurrent neural networks. In *International Conference on Engineering Applications of Neural Networks*, pages 229–239. Springer, 2018.

[25] Nikita Dvornik, Cordelia Schmid, and Julien Mairal. Selecting relevant features from a multi-domain representation for few-shot classification. In *European Conference on Computer Vision*, pages 769–786. Springer, 2020.

[26] Tamer Elsayed, Jimmy Lin, and Douglas W Oard. Pairwise document similarity in large collections with mapreduce. In *Proceedings of ACL-08: HLT, Short Papers*, pages 265–268, 2008.

[27] Ayesha Enayet and Gita Sukthankar. A transfer learning approach for dialogue act classification of GitHub issue comments. *CoRR*, abs/2011.04867, 2020.

[28] Ayesha Enayet and Gita Sukthankar. Analyzing team performance with embeddings from multiparty dialogues. In *IEEE International Conference on Semantic Computing (ICSC)*, pages 33–39, 2021.

[29] Ayesha Enayet and Gita Sukthankar. Learning a generalizable model of team conflict from multiparty dialogues. *International Journal of Semantic Computing*, 15(04):441–460, 2021.

[30] Aysu Ezen-Can, Joseph F Grafsgaard, James C Lester, and Kristy Elizabeth Boyer. Classifying student dialogue acts with multimodal learning analytics. In *Proceedings of the Fifth International Conference on Learning Analytics and Knowledge*, pages 280–289, 2015.

[31] Nikolaos Flemotomos, Benjamin Ma, and Raghuveer Peri. Coordination or dominance? an investigation of social dynamics in conversational entrainment. 2021.

[32] Jamie Fraser, Ioannis Papaioannou, and Oliver Lemon. Spoken conversational ai in video games: Emotional dialogue management increases user engagement. In *Proceedings of the 18th International Conference on Intelligent Virtual Agents*, pages 179–184, 2018.

[33] Alexander Frummet, David Elsweiler, and Bernd Ludwig. Detecting domain-specific information needs in conversational search dialogues. 2019.

[34] Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. Samsum corpus: A human-annotated dialogue dataset for abstractive summarization. *arXiv preprint arXiv:1911.12237*, 2019.

[35] Chih-Wen Goo and Yun-Nung Chen. Abstractive dialogue summarization with sentence-gated modeling optimized by dialogue acts. In *IEEE Spoken Language Technology Workshop (SLT)*, pages 735–742, 2018.

[36] Sergio Grau, Emilio Sanchis, Maria Jose Castro, and David Vilar. Dialogue act classification using a bayesian approach. In *9th Conference Speech and Computer*, 2004.

[37] David Griol, Javier Carbó, and José M Molina. An automatic dialog simulation technique to develop and evaluate interactive conversational agents. *Applied Artificial Intelligence*, 27 (9):759–780, 2013.

[38] Emitza Guzman, David Azócar, and Yang Li. Sentiment analysis of commit comments in GitHub: An empirical study. In *Proceedings of the Working Conference on Mining Software Repositories*, page 352–355, 2014.

[39] Tobias Heidenreich, Jakob-Moritz Eberl, Fabienne Lind, and Hajo Boomgaarden. Political migration discourses on social media: a comparative perspective on visibility and sentiment across political facebook accounts in europe. *Journal of Ethnic and Migration Studies*, 46 (7):1261–1280, 2020.

[40] Lixiao Huang, Jared Freeman, Nancy Cooke, Samantha Dubrow, John "JCR" Colonna-Romano, Matt Wood, Verica Buchanan, Stephen Caufman, and Xiaoyun Yin. Artificial Social Intelligence for Successful Teams (ASIST) Study 2, 2022.

[41] Md Rakibul Islam and Minhaz F. Zibran. *Towards understanding and exploiting developers' emotional variations in software engineering*. 2016. doi: 10.1109/sera.2016.7516145.

[42] Muhammad Abdullah Jamal and Guo-Jun Qi. Task agnostic meta-learning for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11719–11727, 2019.

[43] K.A. Jehn. A multimethod examination of the benefits and determinants of intragroup conflict. *Administrative Science Quarterly*, 40:256–282, 1995.

[44] Karen A Jehn and Elizabeth A Mannix. The dynamic nature of conflict: A longitudinal study of intragroup conflict and group performance. *Academy of Management Journal*, 44 (2):238–251, 2001.

[45] Jing Jiang and ChengXiang Zhai. Instance weighting for domain adaptation in NLP. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics*, pages 264–271, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[46] Malte F Jung. Coupling interactions and performance: Predicting team performance from thin slices of conflict. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 23(3): 1–32, 2016.

[47] Daniel Jurafsky, Elizabeth Shriberg, and Debra Biasca. Switchboard SWBD-DAMSL shallow-discourse-function annotation coders manual, draft 13. Technical Report 97-02, University of Colorado, Boulder Institute of Cognitive Science, Boulder, CO, 1997.

[48] Anup K Kalia, Norbou Buchler, Arwen DeCostanza, and Munindar P Singh. Computing team process measures from the structure and content of broadcast collaborative communications. *IEEE Transactions on Computational Social Systems*, 4(2):26–39, 2017.

[49] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8420–8429, 2019.

[50] R. Kikas, M. Dumas, and D. Pfahl. Using dynamic and contextual features to predict issue lifetime in GitHub projects. In *IEEE/ACM Working Conference on Mining Software Repositories (MSR)*, pages 291–302, 2016.

[51] Jedrzej Kozerawski and Matthew Turk. One-class meta-learning: Towards generalizable few-shot open-set classification. *arXiv preprint arXiv:2109.06859*, 2021.

[52] Uliyana Kubasova, Gabriel Murray, and McKenzie Braley. Analyzing verbal and nonverbal features for predicting group performance. *arXiv preprint arXiv:1907.01369*, 2019.

[53] Rahul Kulkarni, Kevin Hanna, and Justin Stanely. NLP generalization for QA tasks. 2020.

[54] Harshit Kumar, Arvind Agarwal, and Sachindra Joshi. Dialogue-act-driven conversation model: An experimental study. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1246–1256, 2018.

[55] Catherine Lai and Gabriel Murray. Predicting group satisfaction in meeting discussions. In *Proceedings of the Workshop on Modeling Cognitive Processes from Multimodal Data*, pages 1–8, 2018.

[56] Ofer Lavi, Ella Rabinovich, Segev Shlomov, David Boaz, Inbal Ronen, and Ateret Anaby Tavor. We've had this conversation before: A novel approach to measuring dialog similarity. In *Proceedings of the Conference on Empirical Methods in Natural Language*

Processing, pages 1169–1177. Association for Computational Linguistics, 2021. URL https://aclanthology.org/2021.emnlp-main.89.

[57] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1188–1196, Bejing, China, 22–24 Jun 2014. PMLR.

[58] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196. PMLR, 2014.

[59] Cheongjae Lee, Sangkeun Jung, Jihyun Eun, Minwoo Jeong, and Gary Geunbae Lee. A situation-based dialogue management using dialogue examples. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 1, pages I–I. IEEE, 2006.

[60] Ruizhe Li, Chenghua Lin, Matthew Collinson, Xiao Li, and Guanyi Chen. A dual-attention hierarchical recurrent neural network for dialogue act classification. *arXiv preprint arXiv:1810.09154*, 2018.

[61] Diane Litman, Susannah Paletz, Zahra Rahimi, Stefani Allegretti, and Caitlin Rice. The Teams corpus and entrainment in multi-party spoken dialogues. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1421–1431, 2016.

[62] Qian Liu, Heyan Huang, Jie Lut, Yang Gao, and Guangquan Zhang. Enhanced word embedding similarity measures using fuzzy rules for query expansion. In *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–6. IEEE, 2017.

[63] Sa Liu, Lin Wang, Sien Lin, Zhi Yang, and Xiaofan Wang. Analysis and prediction of team

performance based on interaction networks. In *Chinese Control Conference (CCC)*, pages 11250–11255. IEEE, 2017.

[64] Nichola Lubold. *Producing acoustic-prosodic entrainment in a robotic learning companion to build learner rapport*. PhD thesis, Arizona State University, 2018.

[65] Ellyn Maese, Pablo Diego-Rosell, Les DeBusk-Lane, and Nathan Kress. Development of emergent leadership measurement: Implications for human-machine teams. In *AAAI Symposium on Computational Theory of Mind for Human-Machine Teams*, 2021.

[66] Kamaran H Manguri, Rebaz N Ramadhan, and Pshko R Mohammed Amin. Twitter sentiment analysis on worldwide covid-19 outbreaks. *Kurdistan Journal of Applied Research*, pages 54–65, 2020.

[67] Shannon Marlow, Christina Lacerenza, Jensine Paoletti, C. Shawn Burke, and Eduardo Salas. Does team communication represent a one-size-fits-all approach?: A meta-analysis of team communication and performance. *Organization Behavior and Human Decision Processes*, 144:145–170, 2018.

[68] J. E. McGrath. Time, interaction, and performance. *Small Group Research*, 1991.

[69] T Daniel Midgley and Cara MacNish. Automatic dialogue segmentation using discourse chunking. In *Australasian Joint Conference on Artificial Intelligence*, pages 772–782. Springer, 2003.

[70] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

[71] Dmitrijs Milajevs and Matthew Purver. Investigating the contribution of distributional se-

mantic information for dialogue act classification. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, pages 40–47, 2014.

[72] Pierrick Milhorat, Divesh Lala, Koji Inoue, Tianyu Zhao, Masanari Ishida, Katsuya Takanashi, Shizuka Nakamura, and Tatsuya Kawahara. A conversational dialogue manager for the humanoid robot erica. In *Advanced Social Interaction with Agents*, pages 119–131. Springer, 2019.

[73] John Miller, Karl Krauth, Benjamin Recht, and Ludwig Schmidt. The effect of natural distribution shift on question answering models. In *International Conference on Machine Learning*, pages 6905–6916. PMLR, 2020.

[74] Masahiro Mizukami, Koichiro Yoshino, Graham Neubig, David Traum, and Satoshi Nakamura. Analyzing the effect of entrainment on dialogue acts. In *Proceedings of the Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 310–318, Los Angeles, September 2016. Association for Computational Linguistics.

[75] César Montenegro, Asier López Zorrilla, Javier Mikel Olaso, Roberto Santana, Raquel Justo, Jose A Lozano, and María Inés Torres. A dialogue-act taxonomy for a virtual coach designed to improve the life of elderly. *Multimodal Technologies and Interaction*, 3(3):52, 2019.

[76] John X Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. *arXiv preprint arXiv:2005.05909*, 2020.

[77] Alessandro Murgia, Marco Ortu, Parastou Tourani, Bram Adams, and Serge Demeyer. An exploratory qualitative and quantitative analysis of emotions in issue report comments of open source systems. *Empirical Software Engineering*, 23(1):521–564, 2018. ISSN 1382-3256. doi: 10.1007/s10664-017-9526-0.

[78] Gabriel Murray and Catharine Oertel. Predicting group performance in task-based interaction. In *Proceedings of the ACM International Conference on Multimodal Interaction*, pages 14–20, 2018.

[79] Muhammad Zidny Naf'an, Alhamda Adisoka Bimantara, Afiatari Larasati, Ezar Mega Risondang, and Novanda Alim Setya Nugraha. Sentiment analysis of cyberbullying on instagram user comments. *Journal of Data Science and Its Applications*, 2(1):38–48, 2019.

[80] Usman Naseem, Imran Razzak, Katarzyna Musial, and Muhammad Imran. Transformer based deep intelligent contextual embedding for twitter sentiment analysis. *Future Generation Computer Systems*, 113:58–69, 2020.

[81] Mazni Omar, Sharifah-Lailee Syed-Abdullah, and Naimah Mohd Hussin. Developing a team performance prediction model: A rough sets approach. In *International Conference on Informatics Engineering and Information Science*, pages 691–705. Springer, 2011.

[82] Marco Ortu, Tracy Hall, Michele Marchesi, Roberto Tonelli, David Bowes, and Giuseppe Destefanis. Mining communication patterns in software development: A GitHub analysis. In *International Conference on Predictive Models and Data Analytics in Software Engineering*, 2018.

[83] Jennifer S Pardo. On phonetic convergence during conversational interaction. *The Journal of the Acoustical Society of America*, 119(4):2382—2393, 2006.

[84] Kyungjin Park, Hyunwoo Sohn, Wookhee Min, Bradford Mott, Krista Glazewski, C Hmelo-Silver, and James Lester. Disruptive talk detection in multi-party dialogue within collaborative learning environments with a regularized user-aware network. In *Proceedings of the Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 2022.

[85] Simon Parsons, Steven Poltrock, Helen Bowyer, and Yuqing Tang. Analysis of a recorded

team coordination dialogue. In *Proceedings of the Second Annual Conference of the ITA*, 2008.

[86] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[87] Flora Poecze, Claus Ebster, and Christine Strauss. Social media metrics and sentiment analysis to evaluate the effectiveness of social media posts. *Procedia computer science*, 130:660–666, 2018.

[88] Samuel L Pugh, Arjun Rao, Angela EB Stewart, and Sidney K D'Mello. Do speech-based collaboration analytics generalize across task contexts? In *International Learning Analytics and Knowledge Conference*, pages 208–218, 2022.

[89] Zahra Rahimi and Diane Litman. Entrainment2vec: Embedding entrainment for multi-party dialogues. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8681–8688, 2020.

[90] Sujith Ravi and Jihie Kim. Profiling student interactions in threaded discussions with speech act classifiers. *Frontiers in Artificial Intelligence and Applications*, 158:357, 2007.

[91] David Reitter and Johanna D Moore. Predicting success in dialogue. *Proceedings of the ACL*, 2007.

[92] Masoud Reyhani Hamedani and Sang-Wook Kim. On investigating both effectiveness and efficiency of embedding methods in task of similarity computation of nodes in graphs. *Applied Sciences*, 11(1):162, 2021.

[93] Axel Rodriguez, Carlos Argueta, and Yi-Ling Chen. Automatic detection of hate speech on facebook using sentiment and emotion analysis. In *2019 International Conference on*

Artificial Intelligence in Information and Communication (ICAIIC), pages 169–174. IEEE, 2019.

[94] Renata Lopes Rosa, Gisele Maria Schwartz, Wilson Vicente Ruggiero, and Demóstenes Zegarra Rodríguez. A knowledge-based recommendation system that includes sentiment analysis and deep learning. *IEEE Transactions on Industrial Informatics*, 15(4):2124–2135, 2018.

[95] James Owen Ryan, Michael Mateas, and Noah Wardrip-Fruin. A lightweight videogame dialogue manager. In *DiGRA/FDG*, 2016.

[96] Tulika Saha, Sriparna Saha, and Pushpak Bhattacharyya. Tweet act classification: A deep learning based classifier for recognizing speech acts in twitter. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

[97] Tonmoy Saikia, Thomas Brox, and Cordelia Schmid. Optimized generic feature learning for few-shot classification across domains. *arXiv preprint arXiv:2001.07926*, 2020.

[98] Nabizath Saleena et al. An ensemble classification system for twitter sentiment analysis. *Procedia computer science*, 132:937–946, 2018.

[99] Rodrigo Sandoval-Almazan and David Valle-Cruz. Facebook impact and sentiment analysis on political campaigns. In *Proceedings of the 19th annual international conference on digital government research: governance in the data age*, pages 1–7, 2018.

[100] Anna Sauer, Shima Asaadi, and Fabian Küch. Knowledge distillation meets few-shot learning: An approach for few-shot intent classification within and across domains. In *Proceedings of the 4th Workshop on NLP for Conversational AI*, pages 108–119, 2022.

[101] Kartik Sawhney, Marcella Cindy Prasetio, and Suvadip Paul. Community detection using graph structure and semantic understanding of text. *SNAP Stanford University*, 2017.

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

[103] Mattia Segù, Alessio Tonioni, and Federico Tombari. Batch normalization embeddings for deep domain generalization. *arXiv preprint arXiv:2011.12672*, 2020.

[104] Priyanka Sen and Amir Saffari. What do models learn from question answering datasets? *arXiv preprint arXiv:2004.03490*, 2020.

[105] Riccardo Serafin and Barbara Di Eugenio. Flsa: Extending latent semantic analysis with features for dialogue act classification. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 692–699, 2004.

[106] Antonette Shibani, Elizabeth Koh, Vivian Lai, and Kyong Jin Shim. Assessing the language of chat for teamwork dialogue. *Journal of Educational Technology & Society*, 20(2):224–237, 2017.

[107] Babak Maleki Shoja and Nasseh Tabrizi. Customer reviews analysis with deep neural networks for e-commerce recommender systems. *IEEE access*, 7:119121–119130, 2019.

[108] Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–373, 2000.

[109] Hanif Sudira, Alifiannisa Lawami Diar, and Yova Ruldeviyani. Instagram sentiment analysis with naive bayes and knn: exploring customer satisfaction of digital payment services in

indonesia. In *2019 International Workshop on Big Data and Information Security (IWBIS)*, pages 21–26. IEEE, 2019.

[110] Gita Sukthankar, Katia Sycara, Joseph Andrew Giampapa, Christopher Burnett, and Alun Preece. An analysis of salient communications for agent support of human teams. In Virginia Dignum, editor, *Multi-agent Systems: Semantics and Dynamics of Organizational Models*, pages 284–312. IGI Global, 2009.

[111] Lihua Sun, Junpeng Guo, and Yanlin Zhu. Applying uncertainty theory into the restaurant recommender system based on sentiment analysis of online chinese reviews. *World Wide Web*, 22(1):83–100, 2019.

[112] Xiao Sun, Chen Zhang, and Lian Li. Dynamic emotion modelling and anomaly detection in conversation based on emotional transition tensor. *Information Fusion*, 46:11–22, 2019.

[113] Amanuel Tekleab, Narda Quigley, and Paul Tesluk. A longitudinal study of team conflict, conflict management, cohesion, and team effectiveness. *Group and Organization Management*, 34(2):170–205, 2009.

[114] textblob. Textblob. https://textblob.readthedocs.io/en/dev/.

[115] Quan Hung Tran, Gholamreza Haffari, and Ingrid Zukerman. A generative attentional neural network model for dialogue act classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 524–529, 2017.

[116] Quan Hung Tran, Ingrid Zukerman, and Gholamreza Haffari. Preserving distributional information in dialogue act classification. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2151–2156, 2017.

[117] Eleni Triantafillou, Hugo Larochelle, Richard Zemel, and Vincent Dumoulin. Learning a universal template for few-shot dataset generalization. In *International Conference on Machine Learning*, pages 10424–10433. PMLR, 2021.

[118] Stefan Ultes, Lina M Rojas Barahona, Pei-Hao Su, David Vandyke, Dongho Kim, Inigo Casanueva, Paweł Budzianowski, Nikola Mrkšić, Tsung-Hsien Wen, Milica Gasic, et al. Pydial: A multi-domain statistical dialogue system toolkit. In *Proceedings of ACL 2017, System Demonstrations*, pages 73–78, 2017.

[119] D. N. Walton and E. C. W. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. State University of New York Press, 1995.

[120] Jiancheng Wang, Jingjing Wang, Changlong Sun, Shoushan Li, Xiaozhong Liu, Luo Si, Min Zhang, and Guodong Zhou. Sentiment classification in customer service dialogue with topic-aware multi-task learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9177–9184, 2020.

[121] Lei Wang, Jianwei Niu, and Shui Yu. Sentidiff: combining textual information and sentiment diffusion patterns for twitter sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, 32(10):2026–2039, 2019.

[122] Shujun Wang, Lequan Yu, Kang Li, Xin Yang, Chi-Wing Fu, and Pheng-Ann Heng. Dofe: Domain-oriented feature embedding for generalizable fundus image segmentation on unseen datasets. *IEEE Transactions on Medical Imaging*, 39(12):4237–4248, 2020.

[123] X. Wang, M. Lee, A. Pinchbeck, and F. Fard. Where does LDA sit for GitHub? In *IEEE/ACM International Conference on Automated Software Engineering Workshop (ASEW)*, pages 94–97, 2019.

[124] Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni. Generalizing from a few

examples: A survey on few-shot learning. *ACM Comput. Surv.*, 53(3), jun 2020. ISSN 0360-0300.

[125] Yibo Wang, Mingming Wang, and Wei Xu. A sentiment-enhanced hybrid recommender system for movie recommendation: a big data analytics framework. *Wireless Communications and Mobile Computing*, 2018, 2018.

[126] Nick Webb, Mark Hepple, and Yorick Wilks. Dialogue act classification based on intra-utterance features. In *Proceedings of the AAAI Workshop on Spoken Language Understanding*, volume 4, page 5. Citeseer, 2005.

[127] Jason Wei and Kai Zou. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6383–6389, Hong Kong, China, November 2019. Association for Computational Linguistics.

[128] Kaiquan Xu, Jiexun Li, and Stephen Shaoyi Liao. Sentiment community detection in social networks. In *Proceedings of the iConference*, pages 804–805. 2011.

[129] B. Yang, X. Wei, and C. Liu. Sentiments analysis in GitHub repositories: An empirical study. In *Asia-Pacific Software Engineering Conference Workshops (APSECW)*, pages 84–89, 2017.

[130] Feng-Sueng Yang and Chen-Huei Chou. Prediction of team performance and members' interaction: A study using neural network. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 290–300. Springer, 2014.

[131] Ying Yang, Grace Njeri Kuria, and Dong-Xiao Gu. Mediating role of trust between leader

communication style and subordinate's work outcomes in project teams. *Engineering Management Journal*, 32(3):152–165, 2020.

[132] Dian Yu and Zhou Yu. Midas: A dialog act annotation scheme for open domain human machine spoken conversations. *arXiv preprint arXiv:1908.10023*, 2019.

[133] Yazhou Zhang, Dawei Song, Xiang Li, Peng Zhang, Panpan Wang, Lu Rong, Guangliang Yu, and Bo Wang. A quantum-like multimodal network framework for modeling interaction dynamics in multiparty conversational sentiment analysis. *Information Fusion*, 62:14–31, 2020.

[134] Ran Zhao, Oscar J Romero, and Alex Rudnicky. Sogo: a social intelligent negotiation dialogue system. In *Proceedings of the 18th International Conference on Intelligent Virtual Agents*, pages 239–246, 2018.